Vector ***PathBlazer***™ ***2.0***

User's Manual

Invitrogen™
life technologies

Vector ***PathBlazer 2.0*** User's Manual

Published by:
Invitrogen
7305 Executive Way
Frederick, MD 21704
www.informaxinc.com

Copyright © 2004 Invitrogen. All rights reserved. This book contains proprietary information of Invitrogen. No part of this document, including design, cover design, and icons, may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without prior written agreement from Invitrogen.

The software described in this document is furnished under a license agreement. Invitrogen and its licensors retain all ownership rights to the software programs offered by Invitrogen and related documentation. Use of the software and related documentation is governed by the license agreement accompanying the software and applicable copyright law.

Vector ***PathBlazer*** is a registered trademark of Invitrogen, in the United States and other countries. Logos of Invitrogen are also trademarks registered in the United States and may be registered in other countries. Other product and brand names are trademarks of their respective owners.

Printed in the United States of America

Invitrogen reserves the right to make changes, without notice, both to this publication and to the product it describes. Information concerning products not manufactured or distributed by Invitrogen is provided without warranty or representation of any kind, and Invitrogen will not be liable for any damages.

This version of the *Vector* ***PathBlazer 2.0*** User's Manual was published in March 2004.

Invitrogen/InforMax Technical Support

USA

Phone: 240-379-4240

800-357-3114 (Toll-free, U.S.)

E-mail: techsupport@informaxinc.com

Europe, Middle East, Africa, Asian Pacific

Phone: +44 (0) 141 814 6350

E-mail: euroinformax@invitrogen.com

TABLE OF CONTENTS

Chapter 1
Introduction to Vector PathBlazer 1

Chapter 2
Overview of Vector PathBlazer 5

Chapter 3
Working with Pathways 9

Chapter 4
Importing Data 65

Chapter 5
Drawing Pathways 109

Chapter 6
Automatically Assembling Pathways 133

Chapter 7
Gene Ontologies 153

Chapter 8
Working with Gene Expression Data 165

Appendix A
License Manager 183

Appendix B
DTD For Data Import 191

Appendix C
References 199

Appendix D
Troubleshooting 203

Glossary 209
Index 211

Table of Contents

Chapter 1
Introduction to Vector PathBlazer 1

Overview 1
Getting Started with Vector NTI PathBlazer 1
Manual Purpose 2
Manual Contents 2
System Requirements 2
Using Online Help 3
Contacting Technical Support 3
Conventions Used in this Manual 4

Chapter 2
Overview of Vector PathBlazer 5

Introduction 5
Main Features 5
Vector PathBlazer Database 6
Main Data Types 6
Pre-Loaded Data 7
Gene Ontologies 8
Integration with Vector Xpression 3.1 8

Chapter 3
Working with Pathways 9

Launching PathBlazer Viewer 10
Creating a New Database 10
Backing Up the Database 11
Elements of PathBlazer Viewer 11
Pathway Viewing Area 12
Database Explorer 12
Menu Bar and Toolbars 13
Working with Pathways in the Graphics Window 13
Opening a Pathway 14
Viewing Pathways Graphically 14
Navigating Objects in the Graphics Window 17

Customizing Graphical Properties 19
Viewing Pathways in Text Format 28
Creating Alternate Graphical Views 29
Working with Pathways in the Database Explorer 31
Browsing Pathway Data 31
Naming, Copying, and Deleting Objects 33
Organizing Pathway Data 33
Reversing the Direction of a Reaction 35
Adding Pathways, Reactions, Experiments, and Components to the Graphics Window 36
Annotating Pathways, Components, Experiments, Reactions, and Connectors 37
Annotation Fields for Components, Reactions, and Pathways 39
Annotation Fields for Connectors 44
Merging Components Manually 45
Saving PathBlazer Components, Reactions and Pathways 46
Saving a Pathway or Reaction to the Database or a File 46
Saving Reactions Not Going Through a Pathway 50
Saving a .pw File to the Database 51
Opening Crosslinks to External Databases 52
Searching the Database 53
Finding an Object in a Pathway 53
Searching Objects in the Database and Creating Subsets 54
Search Database by GO Annotation 61
Printing and Saving Images 63
Printing an Image 63
Saving an Image 63

Chapter 4
Importing Data 65

Introduction to Importing Data 65
About Vector PathBlazer Data Import 66
Import Module and Description 66
Root Folder or Source File Dialog Box 67
Merge Option Dialog Box 67
Import Session Monitor 70
PathBlazer Import Buttons 71
PathBlazer Log File 71
Importing KEGG Data 72
KEGG Source Files 72
KEGG Import Logic 73
KEGG Compound File 73
KEGG Enzyme File 75
KEGG Reaction Files 77
KEGG Genome File 78
Instructions for Importing KEGG 79
Importing BIND Data 80
BIND Source Files 81
BIND Import Logic 83
Instructions for Importing BIND 84
Importing BioCyc Data 85
BioCyc Source Files 86
BioCyc Import Logic 86

BioCyc Component Files 87
BioCyc Reaction Files 89
BioCyc Pathways File 90
Instructions for Importing BioCyc Data 92
Importing TransPath Data 93
TransPath Source Files 93
Instructions for Importing TransPath Data 96
Importing DIP Data 97
DIP Source Files and Import Logic 97
Instructions for Importing DIP 99
Importing PPI Data 100
Instructions for Importing User PPI Data 100
Importing Proprietary Data 102
Defining Components 102
Defining Reactions 103
Defining Pathways 104
Instructions for Importing Proprietary Data 105
Pre-Defined URLs 107

Chapter 5
Drawing Pathways 109

Introduction to Drawing Pathways 109
Drawing Tools 110
Drawing a New Pathway 112

Chapter 6
Automatically Assembling Pathways 133

Introduction 133
Pathway Assembly Parameters 134
Specifying Parameters 134
Selecting Components and Reactions 134
Using Component Subsets to Limit Pathway Interactions 135
Limiting the Number of Steps Between Components 136
Specifying Pathway Direction and Interaction Generality 137
Pathway Colors in the Graphics Window 138
Assembling Metabolic Versus Discovery Pathways 138
Adding Stepwise Reactions to Pathways 138
Building Pathways by Selecting Reactions in the Database Explorer 139
Examples of Automatically Assembling Pathways 139
Before You Begin 139
Building a Pathway from a Starting Component 139
Building a Pathway from a Starting Component to an Ending Component 141
Building a Pathway from a Starting Pathway to an Ending Component 143
Building a Pathway Through a Component 145
Adding a Stepwise Reaction 147
Building A Link Between Two Pathways 149
Showing Connections to Data from Other Datasources 150

Chapter 7
Gene Ontologies 153

Introduction to Gene Ontologies 153

Working with Gene Ontology Terms 154
Importing Gene Ontology Terms 154
Viewing Gene Ontology Terms 155
Searching Gene Ontology Terms 156
Manual Annotation of PathBlazer Objects with GO Terms 157
Updating GO Categories 159
Working with Gene Ontology Annotations 159
Importing Gene Ontology Annotations 159
Population of Organism/Subcellular Location Attributes Based on GO Annotations 161
Sample Workflow Using Gene Annotations 162

Chapter 8
Working with Gene Expression Data 165

Introduction to Expression Data Import and Display 165
Interaction Between Vector PathBlazer 2.0 and Vector Xpression 3.1 166
Linking Gene Expression Data to Pathway Components 166
Creating an Template Automatically 166
Importing Expression Data with a Template 168
Editing a Template 170
Importing a Template 171
Mapping Database Links Manually 171
Creating a Tab-Delimited Data File of Expression Values 174
Exchanging Data Between Vector PathBlazer and Vector Xpression. 176
Creating a Template from Vector Xpression 3.1 176
Searching a Vector Xpression Database 176
Opening an Experiment in Vector Xpression 177
Sending Expression Data to PathBlazer 177
Finding Components in PathBlazer 177
Displaying Expression Data on Pathways 178
Default Display Colors for Expression Values 179
Modifying Display Colors for Expression Value Ranges 181

Appendix A
License Manager 183

License Manager Dialog Box 184

Appendix B
DTD For Data Import 191

Appendix C
References 199

General 199
KEGG 199
Description 199
URL 199
References 200
Licensing Information 200
BIND 200
Description 200
URL 200
References 200

Licensing Information 200
BioCyc 200
Description 200
URL 200
Reference 200
Licensing Information 200
Transpath 201
Description 201
URL 201
Reference 201
Licensing Information 201
DIP 201
Description 201
URL 201
References 201
Licensing Information 202
Pre-Loaded Data 202
Metabolic Pathways 202
Signal Transduction Pathways 202
Gene Expression 202
Interaction Generality 202
Appendix D
Troubleshooting 203
General 203
Import 204
Glossary 209
Index 211

CHAPTER

1

INTRODUCTION TO VECTOR PATHBLAZER

Overview

Welcome to *Vector **PathBlazer*** ™ ***2.0,*** part of a family of software packages developed by , Invitrogen™ Bioinformatics, Frederick, Maryland. Other life science applications developed by Invitrogen include *Vector **NTI Advance***™*, Vector **Xpression***™, ***LabShare***™ *for Vector NTI*® and *Vector **NTI***® *for Mac OS X*.

You may not have purchased licenses for all of the modules of the Vector products. If you would like to do so, please contact Invitrogen at the website: http://www.informaxinc.com for more information.

Vector PathBlazer is a desktop solution for managing and analyzing diverse biological pathways and protein-protein interaction data. Public domain data from KEGG, BIND, DIP, TransPath and BioCyc databases as well as PPI and proprietary data can be combined, edited, and organized based on your research objectives enabling the discovery of novel pathways. Vector PathBlazer integrates with other members of the Vector family of products, including *Vector **NTI Advance*** and *Vector **Xpression***, to manage a complete functional genomics workflow.

Getting Started with Vector NTI PathBlazer

- To learn about the *Vector PathBlazer 2.0 User's Manual* structure, review Chapter 1.
- To read a brief overview of the Vector PathBlazer software, review Chapter 2.
- To activate your license for Vector PathBlazer 2.0:
  - Refer to the *Vector PathBlazer 2.0 Installation and Licensing Guide* that you received when purchasing Vector PathBlazer

  or

  - Download the *Vector PathBlazer 2.0 Installation and Licensing Guide* from the Invitrogen/InforMax website, www.informaxinc.com

  or

    - See the *Appendix A* License Manager in this user's manual.
- To start Vector PathBlazer, refer to *Launching PathBlazer Viewer* on page 10.
- To learn various methods of opening and using of Online Help, refer to *Using Online Help* on page 3.

Manual Purpose

The purpose of this manual is to provide you with information and instructions for using Vector PathBlazer to view, build, and analyze pathway and protein-protein interaction data.

Manual Contents

This manual is organized into chapters that provide information about how to use the program and appendixes that provide supporting information.

Chapter 1 (this chapter) contains a brief introduction, system requirements, and conventions used in the manual.

Chapter 2 provides an overview of Vector PathBlazer features.

Chapter 3 describes how to view, manage and work with pathways in PathBlazer Viewer.

Chapter 4 describes how to import public and proprietary data into Vector PathBlazer.

Chapter 5 describes how to draw pathways *de novo* in PathBlazer Viewer.

Chapter 6 describes how to use Vector PathBlazer to suggest new pathways and protein-protein interaction networks from known components and reactions.

Chapter 7 describes gene ontology terms and annotations, and discusses gene ontology import and assignment to PathBlazer objects.

Chapter 8 describes how to overlay gene expression data on the topology of a pathway.

Appendix A describes the License Manager, used to license Vector PathBlazer.

Appendix B includes the Document Type Definition (DTD) for mapping proprietary data to a PathBlazer-formatted XML file for import.

Appendix C contains a list of references to locations and citations where you can obtain more information about key concepts in Vector PathBlazer.

Appendix D contains a list of troubleshooting tips for problems that you might encounter when using Vector PathBlazer.

Glossary contains definitions of terms or phrases used in the context of PathBlazer

System Requirements

Vector PathBlazer is a single user application that can be installed on a PC only. Installation instructions are provided in a separate manual called the *Installing and Licensing Guide for Vector PathBlazer.* The system requirements for Vector PathBlazer are:

- Microsoft Windows:
    - 98 SE (second edition)
    - NT 4.0 Workstation (service pack 6a)
    - 2000
    - ME
    - XP (Professional)

- 140 Mb Hard Disk space (additional space is required to load KEGG, BioCyc, TransPath and DIP)
- 128 Mb RAM
- Microsoft Installer Version 2
- Web browser
  - Internet Explorer 5.x
  - Netscape Navigator 4.x

Note: If you have Microsoft Internet Explorer, you can automatically check your system for compatibility with the Vector PathBlazer system requirements and upgrade it as necessary. To do this, using MS Internet Explorer, go to the **Downloads** page of the Invitrogen/InforMax web site, http://www.informaxinc.com, and follow the instructions. This option is not available using Netscape Navigator.

Using Online Help

In the Online Help for Vector PathBlazer, you will find explanations of the features in Vector PathBlazer, as well as tips to guide you through the program's basic functionality.

In the software, there are several ways to open Online Help:

- Select **Help > Help Topics** from the menu bar. In the Online Help that opens, you can browse through the Table of Contents or the Index, or launch a word search of the Online Help application.
- Press F1 or click the **Help** button from any open dialog box, opening its associated help topic.

If pressing F1 fails to open an Online Help topic, select **Help > Help Topics**, opening Online Help. Proceed with a browse through the Table of Contents or Index or do a word search. Your topic may be in the Help files, but inadvertently not linked to its associated dialog box.Topics may be titled by their function rather than the dialog box name. For example, the New Molecule dialog box associated topic is named “Creating a New Molecule.”

Tips for using Vector PathBlazer online Help:

- Click **Help Topics** to show or hide the Contents, Index, and Search tabs.
- Click **Print** to print the current topic.
- Click **>>** to go to the next topic in a sequence. Click **<<** to go to the previous topic in a sequence.
- When a **See Also** button is present in a topic, click the button to display a list of related topics that you can go directly to.
- Click the green-colored text to jump to a linked topic.

Contacting Technical Support

USA

Phone: 240-379-4240

800-357-3114 (Toll-free, U.S.)

E-mail: techsupport@informaxinc.com

Europe

Phone: +44 186 5784591

For online technical support, send your questions to: techsupporteurope@informaxinc.com

Conventions Used in this Manual

The following table lists conventions that are used to differentiate between regular text and menu commands, keyboard keys, toolbar buttons, dialog box options and text that you type (Table *1.1*).

Convention	*Description*
Bold & Capitalized Command **Bold & Capitalized command > Bold & Capitalized command**	Indicates a menu command Indicates sequential menu commands *Example*: Select **Edit > Copy**
TEXT IN SMALL CAPS	Keyboard key that you press *Example*: Press ENTER
TEXT IN SMALL CAPS + TEXT IN SMALL CAPS	Keyboard keys that you press concurrently *Example*: Press SHIFT + CTRL and then release both.
TEXT IN SMALL CAPS FOLLOWED BY TEXT IN SMALL CAPS	Keyboard keys that you press in sequence *Example*: Press ENTER, then TAB to commit the change
Icon	A button that you click *Example*: Click the **Delete** button ( ) to delete the component.
Bold type	Options that you select in dialog boxes or drop-down menus. Buttons or icons that you click. *Example*: Click the **Add** button.
Italic & bold type	Text that you type *Example*: In the New Subset text box, enter ***Proprietary Proteins***.
Note: **Warning!** **Important!**	Highlights a concept of particular interest or information of which you should be particularly aware. *Example:* **Note:** This concept is used throughout the manual.
Blue text that is underlined *Blue text, italic font cross reference*	Hyperlinked text. The hyperlinks can be URLs to Web sites, or they can be cross references within the user's manual, hyperlinked in the Vector PathBlazer User's Manual pdf for easy reference. *Examples:* www.informaxinc.com *Gene Ontologies* on page 8

Table 1.1 Text conventions used in this manual

CHAPTER

2

OVERVIEW OF VECTOR PATHBLAZER

This chapter provides a summary of how Vector PathBlazer provides a solution for managing pathway and protein-protein interaction data and describes the database and key data types.

Topics in this chapter include:

- *Main Features* on page 5
- *Vector PathBlazer Database* on page 6
- *Pre-Loaded Data* on page 7
- *Gene Ontologies* on page 8
- *Integration with Vector Xpression 3.1* on page 8

Introduction

Biological science has surpassed the stage of cataloging simple parts and is facing the challenge of understanding a system's function*:* the network of processes and interactions through which the individual catalogued parts interact and function. It is through finding important networks among the various parts under normal and disease conditions that the complex regulatory pathways of biology can be understood at a level to effectively modulate them. In this effort it is critical to draw on all available knowledge and arrive at a solution that combines well-known biological facts with new, less-understood areas. Vector PathBlazer can aid in developing testable hypotheses that can be used to extend biological knowledge.

Main Features

The key features of Vector PathBlazer are:

- stores molecular interaction data from proprietary and public data sources
- imports both proprietary data and public data including KEGG, BIND, DIP, TransPath, PPI and BioCyc databases
- stores components, reactions, and pre-assembled pathways separately in a proprietary data model

- draws component, reactions, and pathways de novo
- assembles potential networks across different data sources
- assembles pathways and protein-protein interaction networks interactively in a step-wise manner using query and filter options and displays resulting pathways and networks in a graphical view
- uses Interaction Generality as a measure to enrich for biologically relevant protein-protein interactions
- annotates pathways, reactions, and components
- links to sequence records, sequence analysis tools, and citations in the Vector NTI database and other external databases.
- displays differential gene expression data from microarrays in the context of a pathway
- imports and assigns gene ontology terms and/or annotations to PathBlazer objects
- launches Vector Xpression 3.1 where an expression experiment selected in PathBlazer displays

Vector PathBlazer Database

Vector PathBlazer is a single-user system and is comprised of a database, which stores all the data objects and the relationships between them, and a client, which allows you to import, view, and manipulate the objects stored in the database. The database is located on the same machine where the program is installed in a file with the extension *.mdb*. When you install Vector PathBlazer and start it for the first time, a default database called *PathBlazer_demo_db.mdb* is loaded with a set of pre-loaded data to *C:\VNTI Database\PathwayDB*. The pre-loaded data consists of a set of example data you can use to learn the program. You can add data to the default database and you can also create new databases. Any data you import, create, or modify is saved to the database you specify.

Main Data Types

The Vector PathBlazer database stores four key data types: Pathways, Reactions, Components, and Experiments. Pathways are made up of reactions which are, in turn, made up of components.

Components—are elements of a reaction and can be either an input, output, or both of the reaction. In Vector PathBlazer, components can be any kind of molecule such as protein, DNA, RNA, small molecule, etc. and can also be physical elements such as heat or light. For example, the substrates, products, and enzyme in the first step of the metabolic pathway glycolysis are glucose, ATP, hexokinase, glucose-6-phosphate, and ADP and can each be represented as individual components. Components can either be imported into the database or created de novo and are stored in the database as individual entities. Components can have attributes associated with them such as subcellular localization, chemical formula, type, etc. and can also have alternate names or **synonyms** associated with them. Components are named by a unique, primary name in the database and synonyms can be used as secondary names. Synonyms are especially useful when searching the database and naming components. For example, hexokinase can have the synonym glucokinase associated with it and when a search is performed for glucokinase, hexokinase is returned as the primary object. Furthermore, if a synonym is entered when building a pathway, any reactions that include components that match by synonym are linked together by the pathway building algorithm.

Reactions—are groups of one or more components that undergo a **transformation**. Transformations are biochemical reactions or interactions between components. The types of transformations that can be represented in Vector PathBlazer are:

- normal (forward or reverse)

- interaction (protein-protein interaction)
- activation
- inhibition
- catalysis

Reactions can be of the type characteristically described in metabolic or signal transduction pathways and have a defined direction as well as substrates and products or can be protein-protein interactions, which consist of two interacting proteins without a defined direction. Similar to components, reactions can have attributes associated with them such as cellular localization, formula, type, etc.

Pathways—are one or more sets of reactions linked together through at least one component. Different types of pathways can be modeled in Vector PathBlazer including metabolic and signal transduction pathways. Pathways can also be made up of networks of protein-protein interactions. Similar to components and reactions, pathways can also have attributes associated with them.

Experiments—Gene expression data can be stored in the PathBlazer 2.0 database as Experiment objects. Experiments are composed of expression values obtained from genes that make up Expression Runs. Experiments (expression values) map to PathBlazer database Components upon import. If expression data were sent to PathBlazer directly from Vector Expression Experiments, the objects also retain reference to the original Vector Expression database.

Through the use of these four main data types, you can construct known pathways and use known information about reactions to discover novel pathways and networks.

Pre-Loaded Data

To aid you in learning to use Vector PathBlazer, several different pathways have been entered from the literature and are pre-loaded in the Vector PathBlazer database for your use. The following pathways are pre-loaded in the default database that is installed when Vector PathBlazer is installed:

- Metabolic
  - Gluconeogenesis
  - Glycolysis
  - Pentose phosphate
  - Tricarboxylic Acid (TCA)
- Signal Transduction
  - TNFR
  - Wnt
  - EGF Signaling

The pathways include the associated components linked into the appropriate reactions. See *Appendix C* for references associated with these pathways.

Since all data records from BIND (Biomolecular Interaction Database) are in the public domain, the BIND interaction database is also pre-loaded in the Vector PathBlazer database. The BIND database is loaded as a set of components and reactions in Vector PathBlazer. For more information, see *Chapter 4*.

Finally, a set of expression values from multiple expression runs that map to the gene products of the enzymes in the glycolysis pathway is pre-loaded. The file containing the values is also included in the default database directory *C:\VNTI Database\Pathway DB\DeRisi_glycolysis_exp_import.txt*. This data is used in *Chapter 8* to demonstrate how expression values are mapped on the topology of a pathway.

Gene Ontologies

Vector PathBlazer allows you to import Gene Ontologies (pre-defined classifications of Genes and Targets) that you download and save locally from the Gene Ontology Consortium. For more information, see Chapter 7 *Gene Ontologies*. From PathBlazer, you can assign selected PathBlazer objects to these ontologies, and they become associated as gene annotations.

Integration with Vector Xpression 3.1

Vector PathBlazer 2.0 includes tools for directly accessing expression data in Vector Xpression 3.1. Vector Xpression 3.1 contains tools for sending gene expression data directly to Vector PathBlazer 2.0. For more information, see Chapter 8 *Working with Gene Expression Data*.

CHAPTER

3

WORKING WITH PATHWAYS

This chapter describes PathBlazer Viewer, the main viewer in Vector PathBlazer that is used to view, draw, and manage pathway data in the database. Drawing pathways is described in Chapter 3 *Working with Pathways*.

Topics in this chapter include:

- *Launching PathBlazer Viewer* on page 10
- *Elements of PathBlazer Viewer* on page 11
- *Working with Pathways in the Graphics Window* on page 13
- *Working with Pathways in the Database Explorer* on page 31
- *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37
- *Saving PathBlazer Components, Reactions and Pathways* on page 46
- *Opening Crosslinks to External Databases* on page 52
- *Searching the Database* on page 53
- *Printing and Saving Images* on page 63

Launching PathBlazer Viewer

To launch the PathBlazer Viewer, select **Start > Programs > InforMax 2003> Vector PathBlazer 2 > PathBlazer 2** from your computer's start menu. PathBlazer Viewer opens and initially displays a blank screen with unavailable toolbars until you open a pathway (*Figure 3.1*).

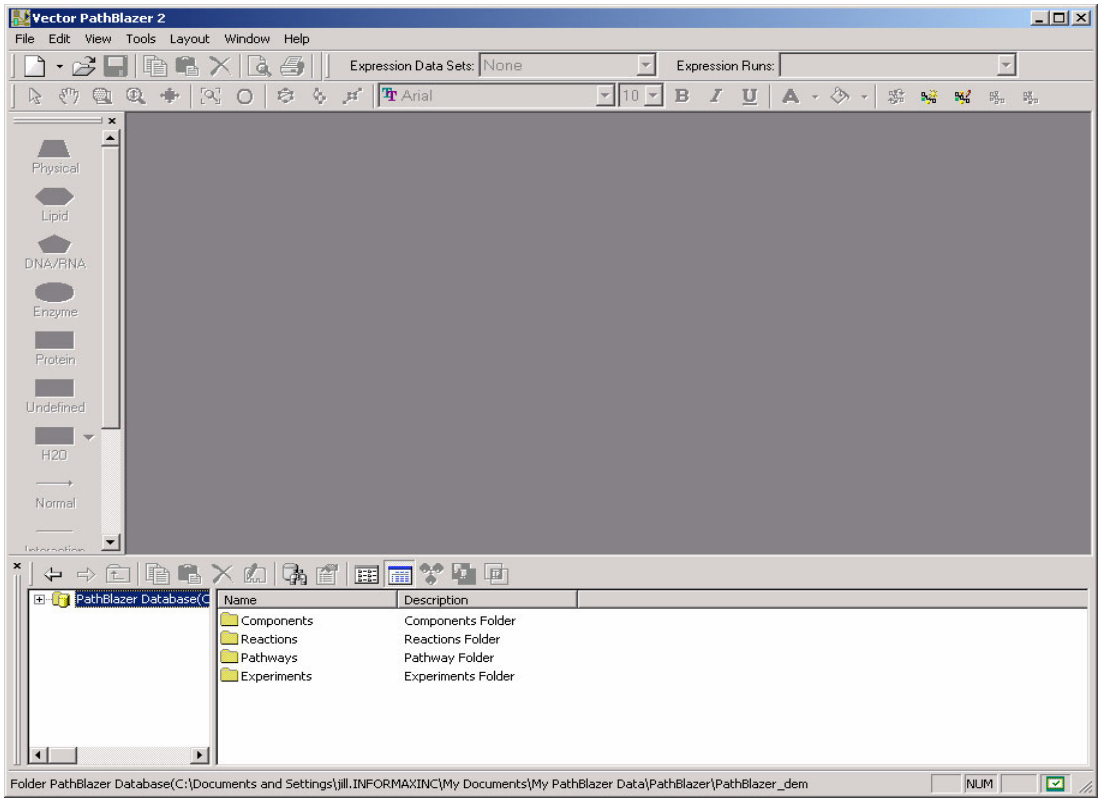


Figure 3.1 PathBlazer Viewer displays a blank screen on initial launch

Creating a New Database

Since Vector PathBlazer is designed for a single user, there are no permission or user identification schemes. All data is visible to any user who starts the program on the computer where it is installed. You can partition individual user data, data sets, projects, experiments, etc. into different databases by creating new databases (that is, new .mbd files) for any of these purposes. However, one of the key features of Vector PathBlazer is the ability to import data from several public sources as well as proprietary data into a single database and use the data together to discover novel pathways and networks. At any time, you can select a new database and view the contents of that database. You can also share individual pathway data by exchanging pathway files (.pw files) with colleagues who also have Vector PathBlazer. For more information about .pw files, see *Saving a .pw File to the Database* on page 51.

***Create a new database*—**by selecting **Tools > Manage Databases > Create New Database**. In the dialog box that opens, enter a name for the database file and navigate to the location where you want to save the file. You can save it to the Vector PathBlazer installation directory or any other directory. Click **Save**. The new database file is created and the path to the file displays as a submenu when you select **Tools > Manage Databases**. The newly created database has a minimal set of important molecules until data is imported into it or created in it.

***Choose a database for use*—**by selecting **Tools > Manage Databases > Select Database**. In the dialog box that opens, navigate to the location of the database file (that is, the .mdb file) you want to use and click **Open**. One database at a time can be viewed in Vector PathBlazer. The paths to recently opened databases display as submenus when you select **Tools > Manage**

Databases. For example, the default database that initially opens when you launch the program may display on the desktop as *C:\My Documents\My PathBlazer Data\PathBlazer_demo_db.mdb*. To view another database, select the appropriate .mdb file.

Important: If you have a PathBlazer 1.0 database, when you choose to open that database in PathBlazer 2.0, you will receive a warning saying that the database will be automatically converted to PathBlazer 2.0 format. Click **OK**. Then be patient, as the database conversion may be somewhat time-consuming.

Backing Up the Database

At certain points in your data collection and annotation process, you may want to take a snapshot of your database or you may want to create backups of one or more databases. Since all of the data is located in one file for a particular database, you can simply copy the associated .mdb file, rename it, and relocate it to an archive or backup directory.

Elements of PathBlazer Viewer

PathBlazer Viewer is the main interface in Vector PathBlazer where pathways are built, viewed, drawn, annotated, and searched.

PathBlazer Viewer is made up of a menu bar and a general toolbar at the top of the window. The Pathway Viewing Area displays in the middle of the window and the Database Explorer and status bar display at the bottom of the window (*Figure 3.2*). The status bar displays the current database settings, such as the number of items currently displayed. It can be hidden from view by selecting **View > Status Bar**. Divider bars separate different areas of the screen and, when the cursor turns to a double-headed arrow, can be dragged to the left or right.

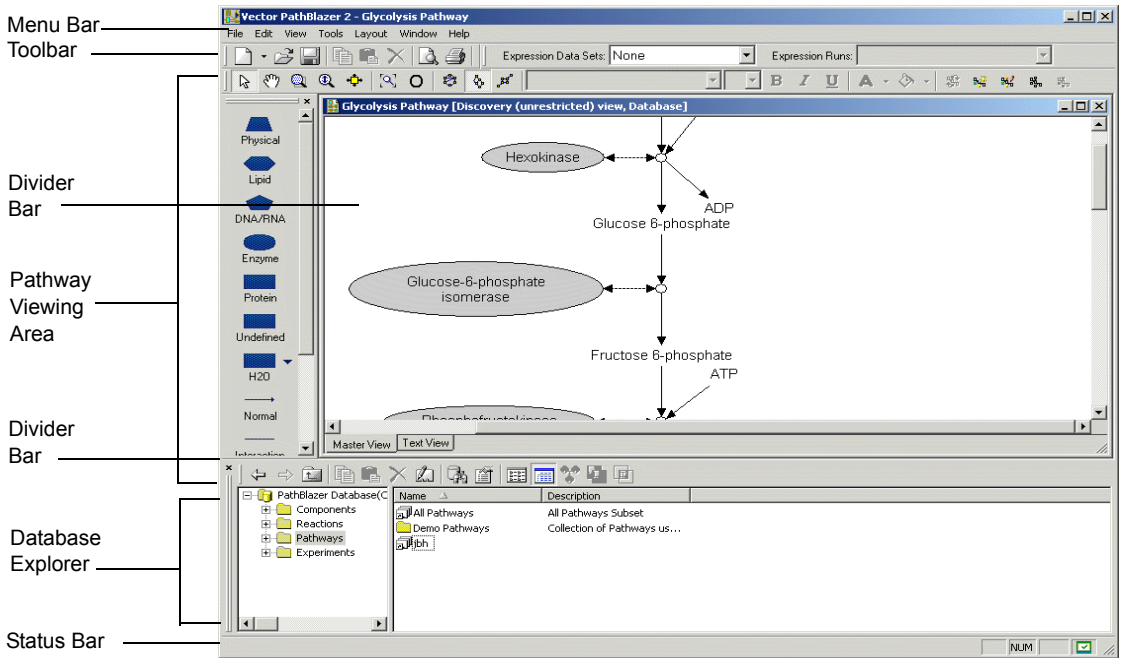


Figure 3.2 Elements of Pathway Viewer

Pathway Viewing Area

The Pathway Viewing Area, in the middle of PathBlazer Viewer, is for building, viewing, drawing, editing, and finding elements in a specified pathway. When you first open PathBlazer Viewer, the Pathway Viewing Area is initially not available until you select a component, reaction, or pathway for display. (Experiments display only in conjunction with an open pathway.)

The Pathway Viewing Area is made up of a Graphics toolbar at the top, a Palette window on the left, and a Graphics window on the right (*Figure 3.3*). The Graphics window initially has two tabs:

1. The **Master View** tab is for viewing the elements of a pathway graphically (see *Viewing Pathways Graphically* on page 14)
2. The **Text View** tab is for viewing the elements in text format (see *Viewing Pathways in Text Format* on page 28).

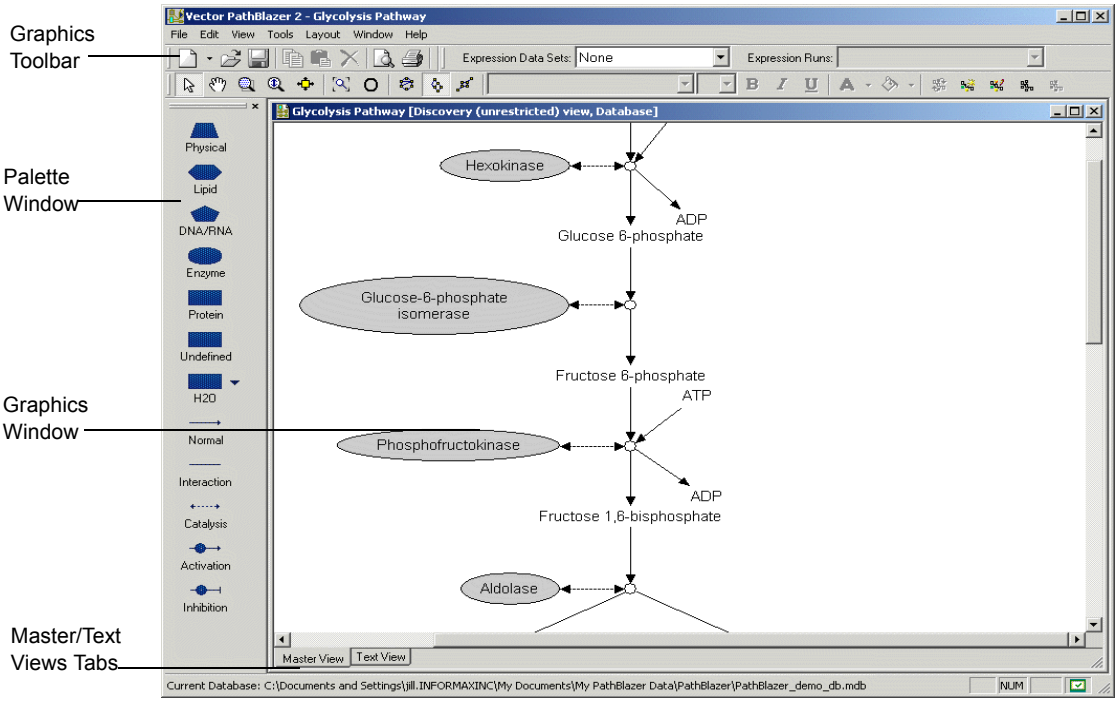


Figure 3.3 Elements of Pathway Viewing Area

The Palette window is anchored on the left side of the screen by default but can be converted to an independent window by dragging on the double-line on the top of the window and dropping it when its borders retract to a smaller rectangle. The window can then be dragged anywhere on the screen. To reanchor it on the left side of the screen again, drag it to the left and drop it when its borders expand to fill the left side or double-click on the its title bar to return it to the left side. For more information about using the Palette, see *Drawing Tools* on page 110.

The Graphics window cannot be converted to an independent window. However, you can maximize the Graphics window by closing the Palette window and the Database Explorer window (described below). To close the Palette window, click the **x** in the right corner. To view the Palette window again, select **View > Palette**.

For more information about using the Pathway Viewing Area, see *Working with Pathways in the Graphics Window* on page 13 and Chapter 5 *Drawing Pathways*.

Database Explorer

The Database Explorer window at the bottom of PathBlazer Viewer (*Figure 3.2*) is for browsing and organizing the contents of the database by the four main data types summarized below. For detailed descriptions of the PathBlazer data types, see *Main Data Types* on page 6.

***Components*—**are elements of a reaction and can be either an input, output, or both of the reaction. They can be any kind of molecule or physical element.

***Reactions*—**are groups of one or more components that undergo biochemical reactions or interactions between components.

***Pathways*—**are one or more reactions; they can be either independent of each other or linked together through at least one component.

***Experiments*—**are expression data whose files are imported into PathBlazer in .xml format.

The Database Explorer window behaves similarly to a Windows-based Explorer and is made up of the Explorer toolbar, the Contents Pane on the left, and the List Pane on the right (*Figure 3.4*). A main folder displays under the **Pathway Database** icon for the four main data types. Selecting a folder or container in the **Contents Pane** displays its contents in the **List Pane**. A divider bar separates the Contents and List Panes and can be dragged to the left or right to change the size of these panes.

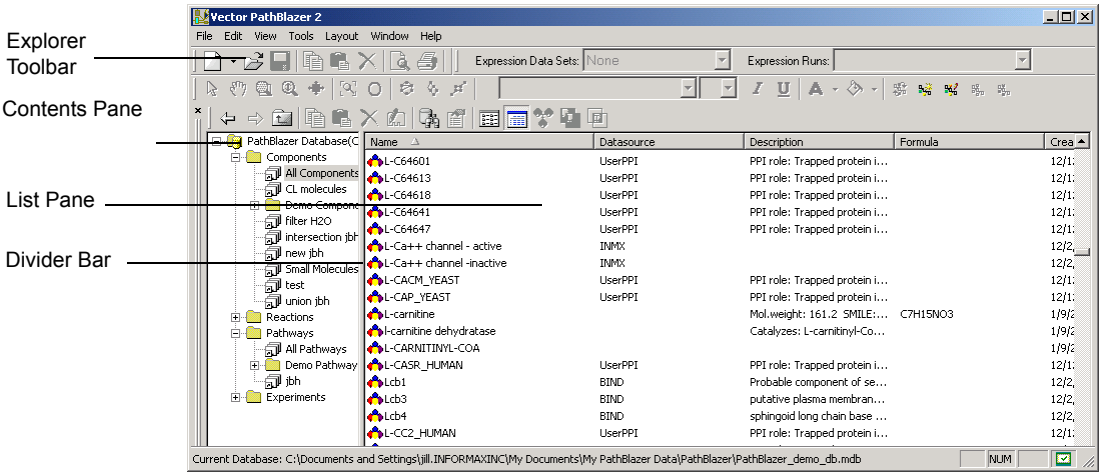


Figure 3.4 Elements of Database Explorer

Similar to the Palette window, the Database Explorer window is anchored at the bottom of the screen by default but can also be converted into an independent window by dragging on the double-line on the far left and dropping the window when it its borders retract to a smaller rectangle. The window can then be dragged anywhere on the screen. To reanchor it at the bottom of the screen again, drag it to the bottom and drop it when its borders expand to fill the bottom or double-click on the its title bar to return it to the bottom. To close the window, click the (**x**) in the upper left corner. To view the Database Explorer window again, select **View > Explorer Pane**.

Menu Bar and Toolbars

Menu commands and toolbar buttons are described throughout this chapter according to their use in the program.

Working with Pathways in the Graphics Window

Before learning how to draw and build pathways, it is important to first understand how pathways, reactions, experiments and components and the relationships between them are represented in Vector PathBlazer. Pathways, the reactions that make up a pathway, and the components that make up a reaction can be graphically or textually displayed in the Graphics window.

Opening a Pathway

Pathways are either stored in the database (that is, an *.mdb* file or an exchangeable XML file having the extension *.pw).* A .pw file is a "mini-database" that stores an individual pathway, its associated reactions and components, and all annotations. These files can be used to share specific pathways with colleagues who also have Vector PathBlazer. See *Saving a .pw File to the Database* on page 51 for instructions on how to save a pathway as a .pw file.

***Open a pathway stored in the database*—**by locating the **Pathways** folder in the Database Explorer and double-clicking on **All Pathways**. In the Contents Pane, all pathways in the database display. Locate a pathway in the **Name** column and double-click on it. A graphical representation displays in the **Master View** tab of the Graphics window. (For information about the **Text View** tab, see *Viewing Pathways in Text Format* on page 28). Initially the elements of the pathway are sized so you can easily see them in the Graphics window (*Figure 3.7*). However, the entire pathway may not visible. Use the scroll bars to see the parts of the pathway that are not immediately visible. In addition to pathways, reactions and components can also be selected in Database Explorer and opened in the Graphics window. For instructions, see *Adding Pathways, Reactions, Experiments, and Components to the Graphics Window* on page 36.

***Open a pathway from a .pw file*—**by selecting **File > Open** or clicking the **Open** button ( ) on the toolbar. In the **Open** dialog box, locate the .pw file and click **Open**. The pathway opens in the Graphics window. You can also launch PathBlazer Viewer and open a pathway at the same time by double-clicking on the .pw file. Paths to recently opened .pw files display at the bottom of the **File** menu.

Note: Since a .pw file only contains information about an individual pathway and its associated reactions and components, any operations such as searching or adding reactions are only performed on the data in the .pw file. For more information about a .pw file, see *Saving PathBlazer Components, Reactions and Pathways* on page 46.

Viewing Pathways Graphically

Components are represented in the Graphics window as either text labels or text labels inside shapes. The display format of a pathway depends on whether it is represented as a discovery pathway or a metabolic pathway.

- In a **Metabolic** pathway, the catalyzing agents (that is, enzymes) are represented as labels unconnected to the assembled pathway. The properties for enzymes thus displayed cannot be accessed from the graphic view in a metabolic pathway.
- In a **Discovery** pathway, enzymes are represented as oval shapes, connected by arrows to the open pathway. Properties for the enzymes can be accessed from the shortcut menu associated with the displayed components.

Components are linked to reactions by **connectors**, which are represented as single or double-headed arrows or straight lines. Generally, reactions are represented as circles or **reaction nodes** and are linked by connectors to the components that are included in a particular reaction. Two kinds of reactions can be represented in Vector PathBlazer: directed reactions and protein-protein interactions. A directed reaction can be represented unidirectionally (that is, forward or reverse) or bidirectionally (that is, forward and reverse) as lines with single or double-headed arrows. The direction of an arrow indicates how a component contributes to a reaction. How-

ever, a directed reaction does not necessarily have to end in a product. Several examples of directed reactions are shown in *Figure 3.5* and *Figure 3.6*.

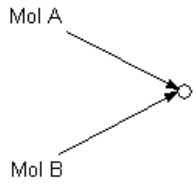


Figure 3.5 Mol A interacts with Mol B (both are Components) in a forward direction (the reaction is represented by a circle) with no known product(s)

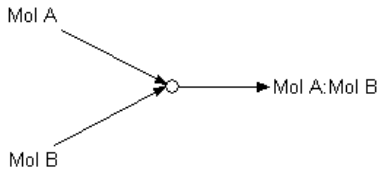


Figure 3.5 Mol A interacts with Mol B in a forward direction to form a complex

A protein-protein interaction involves two proteins interacting without a reaction direction or resulting product. In protein-protein interactions, the reaction node is hidden and the connector between two protein components is represented as a straight line. The following example shows a protein-protein interaction.

Figure 3.6 This protein-protein interaction is the customary way to represent PPI reactions. This example, having no product, is equivalent to the first example in Figure 3.5; these two reactions will have the same representation in the database.

The first reaction in glycolysis is shown in *Figure 3.7*. The entire reaction is outlined with a dotted line to emphasize that a reaction is made up of components, connectors, and a reaction node. The input components in the reaction are glucose and ATP and the output components are glucose-6-phosphate and ADP. The connectors from glucose and ATP to the reaction node point *to* the node, indicating they are substrates. The connectors to glucose-6-phosphate and ADP point *from* the reaction node, indicating these components are products of the reaction. The double-headed arrow between hexokinase and the reaction node indicates that the enzyme catalyzes the reaction.

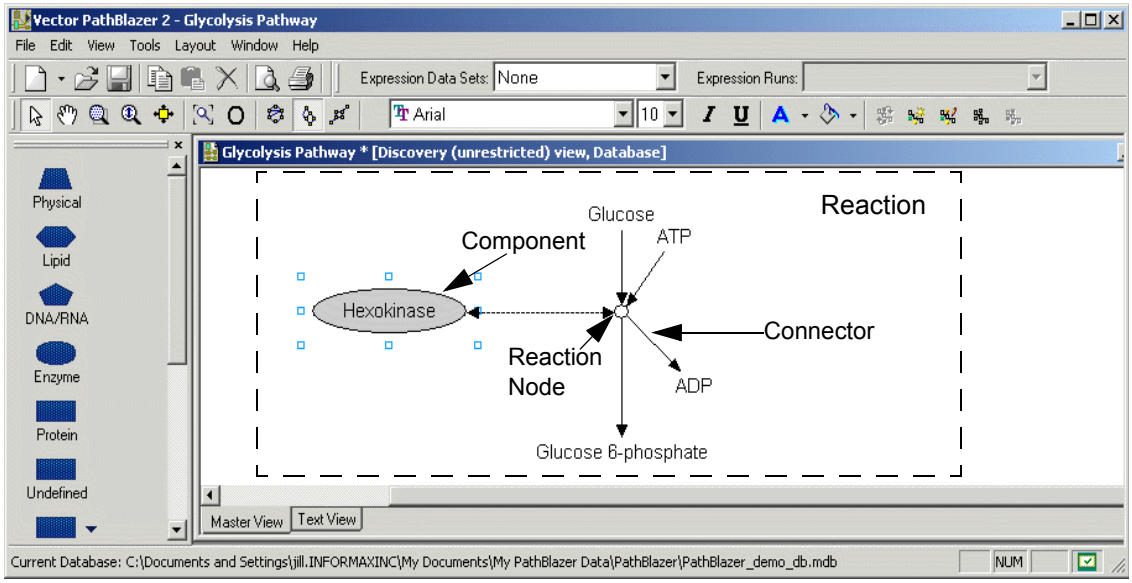


Figure 3.7 Step 1 of glycolysis represented in the Graphics window

Each component, connector, and reaction is drawn independently of other elements in the Graphics window, and each element can be moved independently and has its own graphical properties and physical attributes. The graph itself also has its own graphical properties.

Several additional examples follow showing how components and reactions can be represented in the Graphics Pane.

A+B->C—*Figure 3.8* shows a unidirectional reaction in which two substrates (Mol A and Mol B) react to form one product (Mol C). A separate arrow is drawn from Molecule A and Molecule B to the reaction node in a left to right direction showing that both of these two components (or substrates) are required for the reaction to proceed. A single arrow is drawn from the reaction node to Molecule C in a left to right direction showing that it is the result of the reaction.

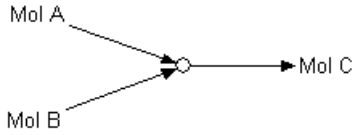


Figure 3.8 Unidirectional reaction with substrates and product

A+B <-> C—*Figure 3.9* shows a bidirectional reaction. In the forward reaction, Mol A and Mol B are substrates and are connected to the 'Forward Rxn' node. Mol C is the product of the forward reaction and is also connected to the 'Forward Rxn' node. All connectors point to the right. The reverse reaction is exactly opposite of the forward reaction. Mol C is now the substrate and is connected to the 'Reverse Rxn' node by a left pointing connector. Mol A and Mol B are the products and are also connected to the 'Reverse Rxn' node by left-pointing arrows.

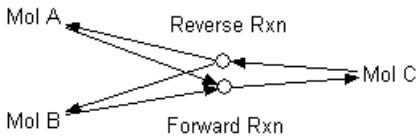


Figure 3.9 Bidirectional reaction with substrates and products

Inhibition and Activation of A+B->C—*Figure 3.10* shows how an inhibiting connector displays as a line a - sign in a circle. An activating connector displays as a line with + sign in a circle.

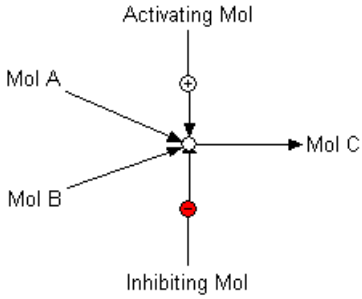


Figure 3.10 Inhibition and activation of a unidirectional reaction

Multimer formation—*Figure 3.11* shows how multimers (dimers, trimers, etc.) can be formed in three separate reactions from Mol A. In Reaction 1, two molecules of Mol A form a dimer called A2. Likewise in Reaction 2, three molecules of Mol A form a trimer called A3 and in Reaction 3, four molecules form a tetramer called A4. In the database, A2, A3, and A4 are each individual components. The connectors that are associated with each reaction can be annotated with stoichiometric constants also. For example, in Reaction 1, the stoichiometric constant can be set to

two since two molecules of Mol A are required to form A2. For more information about annotating connectors with stoichiometric constants, see *Annotation Fields for Connectors* on page 44.

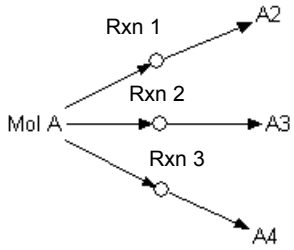


Figure 3.11 Dimer, trimer, and tetramer formation

Navigating Objects in the Graphics Window

You can move, select, and resize individual objects or all objects at the same time in the Graphics window. Use the following operations to move and resize objects.

***Rearrange objects*—**by selecting an object (that is, a component, connector, or reaction node) in the Graphics window and selecting **Tools > Pointer/Select** or the **Arrow** icon ( ) on the Graphics toolbar. Select a component/connector/reaction node and drag it to a new place in the window. When you select a connector ( ) and drag it, a "bend" is introduced ( ). When you select a reaction node and drag it, all of the components and connectors that are linked to it move with it. When you select a component and drag it, any connectors that are linked to it move with it.

***Pan the entire image*—**by selecting **View > Pan** or the **Hand** icon ( ) on the Graphics toolbar. The cursor changes to a hand. As you drag the hand with the mouse in the Graphics window, the entire image in the Graphics window moves with it as one image.

***Resize the image*—**by using one of the following methods.

- Select **View > Overview/Navigation Window** or click the **Overview** button ( ) on the Graphics toolbar to open a second window called the **Overview** window that is independent of the Graphics window (*Figure 3.12*). The Overview window allows you to view the entire pathway while you zoom in on details of the pathway in the Graphics window. The Overview window contains a shaded rectangle or boundary, which can be resized by dragging the handles on any of the corners and can be dragged around the window. As

the boundary is resized and dragged in the Overview window, the contents of the boundary are resized and positioned in the center of the Graphics window.

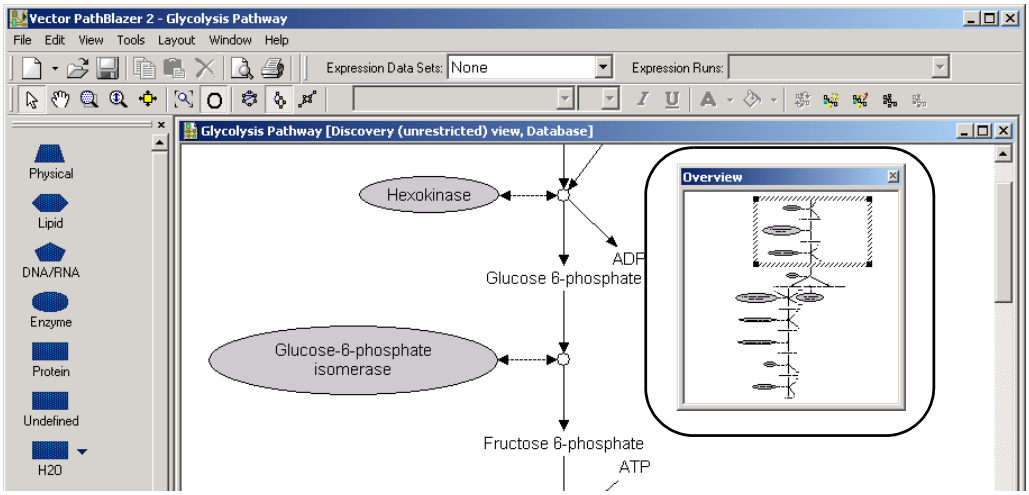


Figure 3.12 Overview window

Note: To tile the Overview window with the Palette window, double-click on the title bar of the Overview window. To return it to an independent window, double-click on its title bar again or drag it from its title bar and drop it anywhere when its width returns to a square.

- Select **View > Zoom** and select one of the following submenus:
  - Select **Fit in window** or click the icon on the Graphics toolbar to "best-fit" all of the pathway elements in the Graphics window.
  - Select a value to zoom to a specified percentage (for example, 400%).
  - Select **Zoom in** or **Zoom out** to zoom in or out. You can also press the **+** or **-** keys to zoom in and out.
  - Select **Marquee Zoom** or click the icon on the Graphics toolbar to change the cursor to a magnifying glass with a crosshair. Drag a wire frame around an area of interest. The area is enlarged when you release the mouse.
  - Select **Interactive Zoom** or click the icon on the Graphics toolbar to change the cursor to a magnifying glass with a two-headed arrow. Drag the mouse vertically and horizontally to zoom in and zoom out on the image.

Jump from connector to the next element (component or reaction node)—by selecting the **Navigate connectors** ( ) button or select **Tools > Navigate Connectors**. The cursor changes to a compass with an arrow pointing out of it ( ). When you point with this icon to a connector, the view jumps to the next component or reaction node. This navigation method is especially useful if you have zoomed in closely on a pathway and want to follow the connectors from component to component.

Multiple select components, reactions, and connectors—by selecting **Edit >**

- **Select All**: to select all objects in the Graphics window.
- **Select All Components/Reactions/Connectors**: to select all components, reactions or connectors.
- **Select All Labels**: to select all labels. Labels are described in *Adding Labels* on page 131.

Selecting all objects of a certain type is especially useful when you want to apply the same graphical properties to them. For more information, see *Customizing Graphical Properties* on page 19.

***Hide components, reactions, and connectors*—**by selecting one or more elements in the Graphics window and selecting **Edit > Hide Selected** or **Hide Selected** from the shortcut menu. A list of submenus displays in the shortcut menu. Select **Hide Selected** to hide only the selected elements, which are hidden from the view with a **+** sign marking their place. To hide multiple levels of elements without selecting them all, select **Hide Children**, **Hide Parents**, or **Hide Neighbors** and then select the number of levels to hide from **One Level**, **N Levels**, or **All Levels**. Unhide selected elements by double-clicking on a **+** sign. If multiple levels are hidden, select **Hide > Unhide Children/Parents/Neighbors**. You can also select **Edit > Unhide All**.

- **Hide Children** hides the resulting elements of the selected element(s) at the selected level. For example, in the reaction on the right side in *Figure 3.13*, if A is selected and **Hide Children > One Level** is selected, then all components, connectors, and reaction node connected to A and including A are replaced with a (**+**) sign.

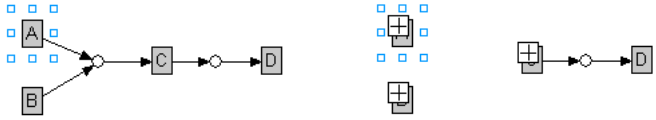


Figure 3.13 Hiding children

- **Hide Parents** hides the forming elements of the selected element at the selected level.
- **Hide Neighbors** hides all elements at the selected level.
- **One Level** means the elements directly associated with the selected element.
- **N Levels** opens a dialog box to enter the number of levels to be hidden that are associated with the selected element.
- **All Levels** means all associated levels of the selected element are hidden.

Customizing Graphical Properties

In the Graphics window, you can customize how individual objects display in terms of shape, size, font, shading, etc. Alternatively, you can apply different graphical layout formats to all like objects in a pathway as a whole, using the customize universally feature. You can also customize display for all objects with a specified gene ontology.

Object and Graph Display Properties

The graphical properties of objects and graphs are those that display in the Graphics window such as the size, shape and color of a component, the font color of a label, and the position of an object in the image as a whole. These properties can be customized for each object.

***View and modify an object's graphical properties*—**by selecting the object in the Graphics window and selecting **View > Object Properties** or **Object Properties** from the shortcut menu. The Object Properties box opens (*Figure 3.14*). Object properties refer to the "node" properties or display properties of an object in the Graphics window including the object's name, its font characteristics, and its shape characteristics. Note that the drop-down list at the top of the win-

dow displays **Selected Node Properties**. This drop-down list toggles to Selected Graph Properties, which are described below.

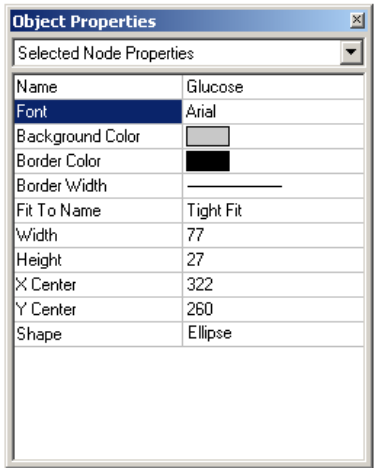


Figure 3.14 Object Properties box for a component

- It is not possible to change an object name in the Object Properties dialog box. The rules for naming an object are listed in *Drawing a New Component* on page 113.
- Change the **Font**, **Background Color**, or **Border Color** by selecting the field and clicking the **Browse** button (...) on the right side of the row. A dialog box for selecting either font characteristics or colors displays. Select the settings you want and click **OK**.
- Change the **Border Width**, **Fit To Name**, and **Shape** by selecting the options from the drop-down list for each. The **Fit To Name** field refers to how the object is shaped when it is resized. Values are:
  - **No fit**: can resize in any direction. Example: ADP > ADP
  - **Tight fit**: cannot resize. If object has been resized, the size reverts to the original default size for the type of object. Example: ADP > ADP
  - **Tight width**: can resize vertically only. Example: ADP > ADP
  - **Tight height**: can resize horizontally only. Example: ADP > ADP
  - **Tight fit preserve aspect**: cannot resize. If object has been resized, the size is retained. Example: ADP > ADP
  - **Preserve aspect ratio**: can scale. If object has been resized, the shape is retained. Example: ADP > ADP
- Change the **Width** and the **Height** by selecting the object in the Graphics window and dragging it by any of the handles. The values for width and height in the Properties box adjust accordingly.
- Change the position of the object from the center of the Graphics window by selecting it and dragging it. The values of **X Center** and **Y Center** adjust accordingly. You can also enter values in these fields in the Properties box and the object moves to the corresponding position in the Graphics window.

Customize an individual object's font and color from the Graphics toolbar—by selecting an object, such as a component, and then changing the font's style, size, and color by making selections from the font buttons and drop-down lists in the Graphics toolbar (*Figure 3.15*). You can also change the fill color of a shape by selecting a color from the drop-down list next to the bucket icon.

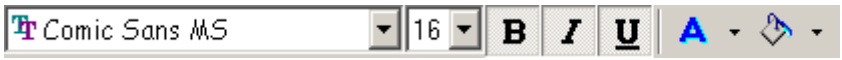


Figure 3.15 Changing object fonts and colors from the Graphics toolbar

Customize universal color schemes and display for selected components and reactions—by selecting **Tools > Filtering/Highlighting > New Filtering/Highlighting Schema**. This opens a Filtering/Highlighting dialog box, with at least **Default Color** listed (*Figure 3.16*).

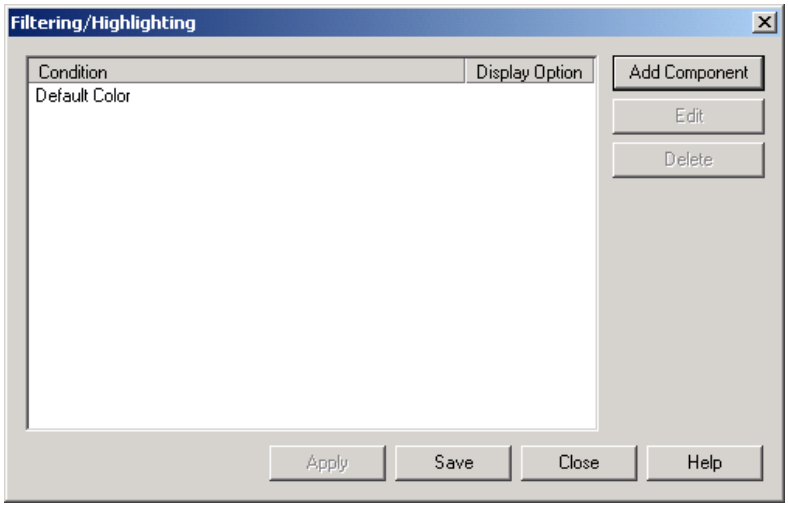


Figure 3.16 The Filtering/Highlighting dialog box allows you to customize display for objects universally

To specify a color for a class of component, click **Add Component**. In the Add Condition dialog box, select the **Condition Type** from the drop-down menu. Then select the **Component Class** from the drop-down menu. When you choose some of the component class options, additional suboptions display (*Figure 3.17*).

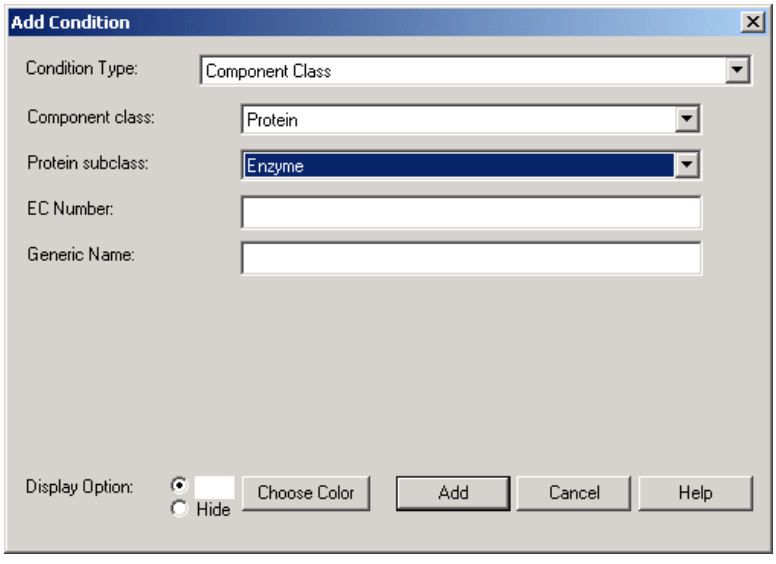


Figure 3.17 The Add Condition dialog box adds suboptions for some of the Condition Type and Component Class selections

Click the **Choose Color** button at the bottom of the box. In the color box that opens, select the color for the specified component display. An alternative is to select the **Hide** radio button, to hide all of the specified components. When you click the **Hide** button a second time, hidden objects display.

As an example, say that you want all enzymes in the open discovery pathway to display with a yellow background. Select **Tools > Filtering/Highlighting > New Filtering/Highlighting Schema**. In the Add Condition dialog box that opens, select **Add Component**. In the Add Condition dialog box, select **Component Class** in the **Condition Type** drop-down menu. In the Component Class drop-down list, select **Protein**. In the **Protein Subclass** drop-down menu, select **Enzyme**. You can enter the **EC Number** (Enzyme Classification #) and **Generic Name** in the appropriate text boxes, but they can be left blank. Click the **Choose Color** button, and select yellow from the color palette. Click **OK**, then click **Add**. That returns you to the Add Condition dialog box where you see the new condition you have just configured (*Figure 3.18*).

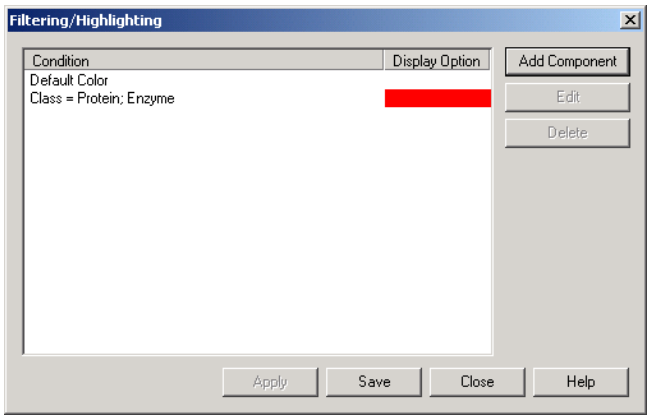


Figure 3.18 The Filtering/Highlighting dialog box displays new conditions for universal display

Click **Apply** to apply the schema to all of the displayed enzymes. To edit any of the conditions listed (including the **Default Color**), select it and click the **Edit** button. To save a schema, select it and click **Save**. Name the schema. Once a schema is saved, it will be listed in the Add Conditions dialog box when you open it. Later you can apply a schema you have saved to any specified objects using the Filtering/Highlighting feature.

Customize gene ontology display—by selecting **Tools > Filtering/Highlighting > New Filtering/Highlighting Schema**. This opens a Filtering/Highlighting dialog box, with **Default Color** listed. To customize gene ontology display, click the **Add Component** button.

Note: Before you can display gene ontologies, you must import the gene ontology files. See *Introduction to Gene Ontologies* on page 153.

In the Add Condition dialog box, select **Component GO Annotation** from the **Condition Type** drop-down menu. This opens an Add Condition dialog box specific to Gene Ontologies. .

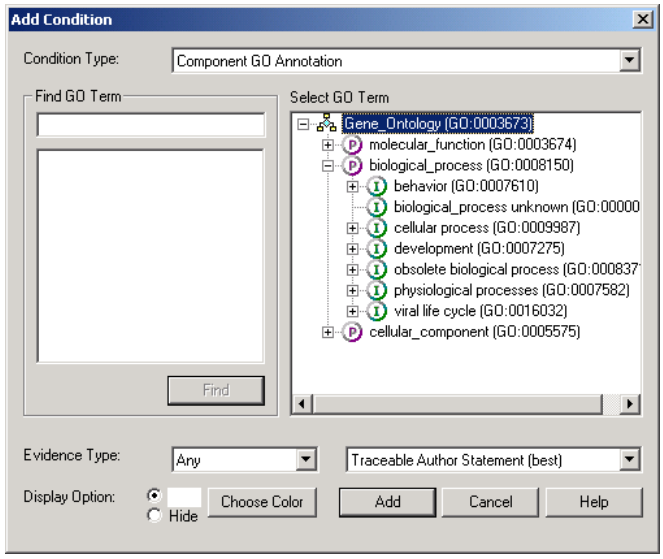


Figure 3.19 Add Condition dialog box for Gene Ontologies

Select any GO term that you would like to set as a condition or search for a GO term by entering the term in the **Find GO Term** text box. Click the **Find** button.

Suboptions that allow you to choose display color or show/hide the annotations display at the bottom of this dialog box.

- Click the **Choose Color** button. In the color box that opens, select the color for the specified GO display. This selection reveals itself in the following way: When a color is applied to a GO term, and the same GO term is associated with a database component, the component displayed in the Graphic Pane exhibits the customized color (*Figure 3.20*).
- Select the **Hide** radio button to hide all of the specified GO annotations. When you click the **Hide** button a second time, hidden objects display.

Once a term and its suboptions are selected, click the **Add** button. The term is added as a condition to the Filtering/Highlighting dialog box. If you select the condition, then click the **Edit** button, you are returned to the Add [GO] Condition dialog box where you can modify your selection. You can also delete the condition by selecting it, then clicking the **Delete** button.

- Click the **Apply** button to execute the filtering/highlighting conditions you have just defined. See *Figure 3.20* to note how a color applied to a GO annotation is implemented.

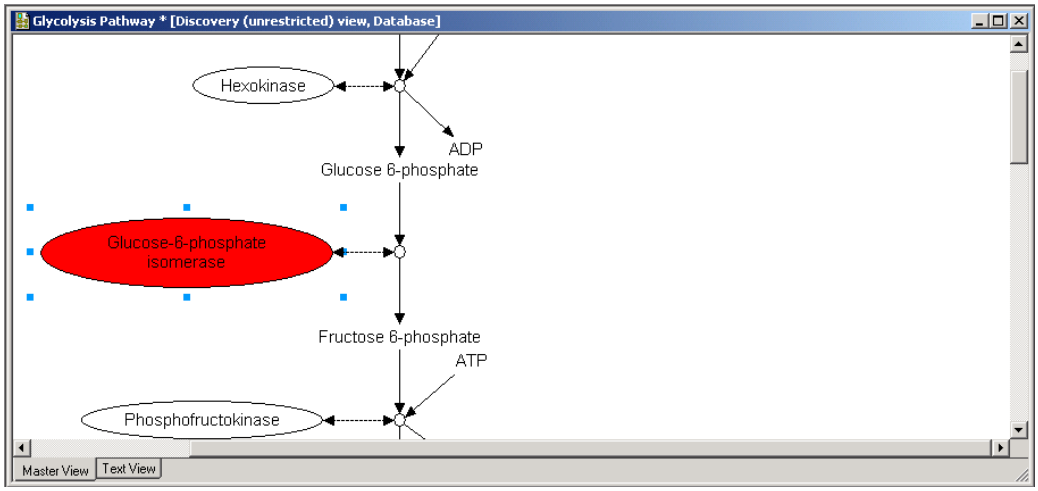


Figure 3.20 When a color is assigned to a GO term that applies to an object in the Graphics window, that object exhibits the color

View and modify a graph's properties—by clicking anywhere, except on an object in the Graphics window, and selecting **View > Object Properties** or **Object Properties** from the shortcut menu. In the Object Properties box that opens, **Selected Graph Properties** displays in the drop-down list at the top (*Figure 3.21*). You can also display the graph's properties when an object's properties are displayed by selecting **Selected Graph Properties** from the drop-down list. The Properties box for a graph summarizes the total number of reaction nodes, connectors, and labels in the pathway. However, the only graph property you can change from this box is the background color. Click the **Browse** button in the **Background Color** field and select a color from the palette.

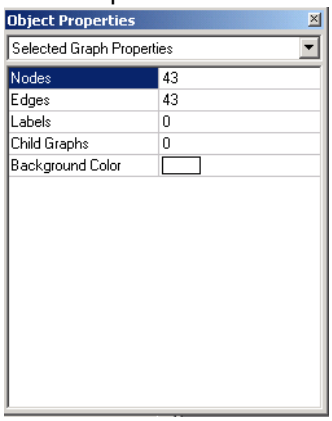


Figure 3.21 Object Properties box for a graph

Graphical Layouts

Graphical layouts are pre-defined orientations that can be applied to a pathway's graphical view. There are three layouts that can be applied to pathways in the Graphics window: Circular, Hierarchical, and Symmetrical. When a layout is applied, the pathway elements are rearranged according to the settings for that layout.

Apply a layout—

- by selecting **Layout > Circular Layout** or clicking the button on the Graphics toolbar.

- by selecting **Layout > Hierarchical Layout** or clicking the button.
- by selecting **Layout > Symmetric Layout** or clicking the button.

Note: When any of these buttons is clicked, the layout converts to a “zoomed out” mode. To zoom in to the graphics, use any of the zoom features described on page 17.

Layout Properties

The parameters in the Layout Properties dialog box determine the settings for each type of layout. Open the Layout Properties dialog box by selecting **Layout > Properties**. Each type of layout corresponds to a tab in this box that contains the layout’s settings.

Circular Tab

The Circular layout settings display in *Figure 3.22* and are described in Table *3.1*.

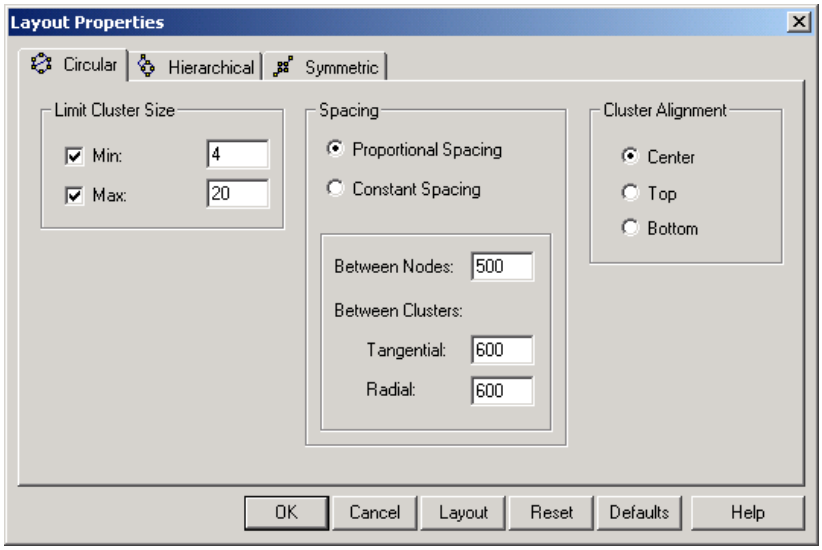


Figure 3.22 Circular Tab of the Layout Properties dialog box

Field	*Settings*	*Description*
Limit Cluster Size	Min Max	Minimum and maximum number of nodes allowed in a cluster. Defaults are Min = 4 and Max = 20.
Spacing	Proportional Spacing	Creates space around nodes proportional to node size. Recommended as the default setting.
	Constant Spacing	Creates space around nodes that is the same for all nodes.
	Between Nodes	Value between nodes.
	Between Clusters Values: Tangential Radial	Dictates spacing between clusters on the circle, around the main cluster, and between the clusters.
Cluster Alignment	Aligns clusters with other clusters’ centers, tops, or bottoms	Alignment tool.

Table 3.1 Settings for Circular layout

Hierarchical Tab

The Hierarchical layout settings display in *Figure 3.23* and are described in Table *3.2*.

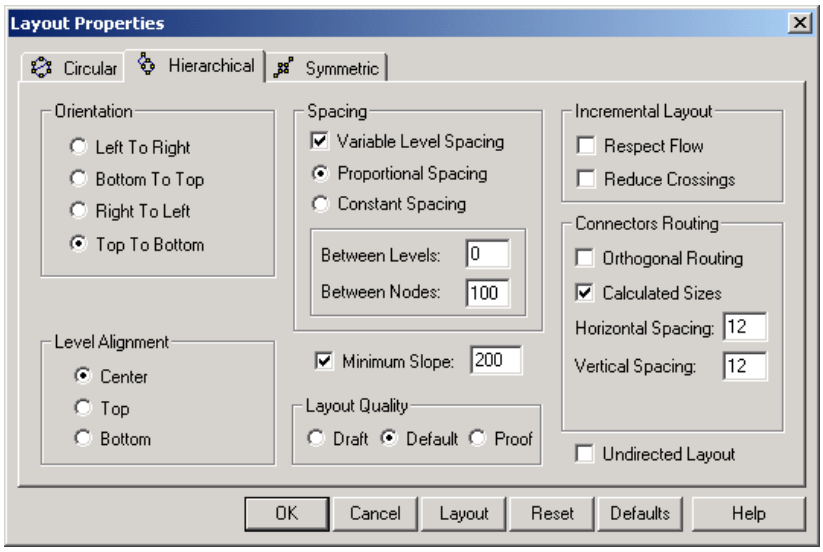


Figure 3.23 Settings for Hierarchical Layout

Field	***Settings***	***Description***
Orientation	Left To Right	Right to left and top to bottom orientation of the image.
	Bottom To Top	
	Right To Left	
	Top To Bottom	
Level Alignment	Center	Aligns nodes.
	Left	
	Right	
Spacing	Variable Level Spacing	Changes the positioning of levels according to the density of edges between levels.
	Proportional Spacing	Creates space around nodes that is proportional to the node size.
	Constant Spacing Values: Between Levels Between Nodes	Creates space around nodes that is the same for all nodes.
Minimum Slope	Checked = On Unchecked = Off	Defines the tangent of an edge slope multiplied by a thousand.
Layout Quality	Draft	Determines how quickly a layout is regraphed and the final quality of a layout.
	Default	
	Proof	

Table 3.2 Settings for Hierarchical layout

Field	Settings	Description
Incremental Layout	Respect Flow	Attempts to place new nodes in the current flow
	Reduce Crossings	Attempts to reduce connector crossings
Connectors Routing	Orthogonal Routing	Turns all connectors to right angles
	Calculated Sizes	
	Horizontal Spacing	
	Vertical Spacing	
Undirected Layout	Checked = On Unchecked = Off	Disregards direction of connectors

Table 3.2 Settings for Hierarchical layout (Continued)

Symmetric Tab

The Symmetric layout settings display in *Figure 3.24* and are described in Table *3.3*.

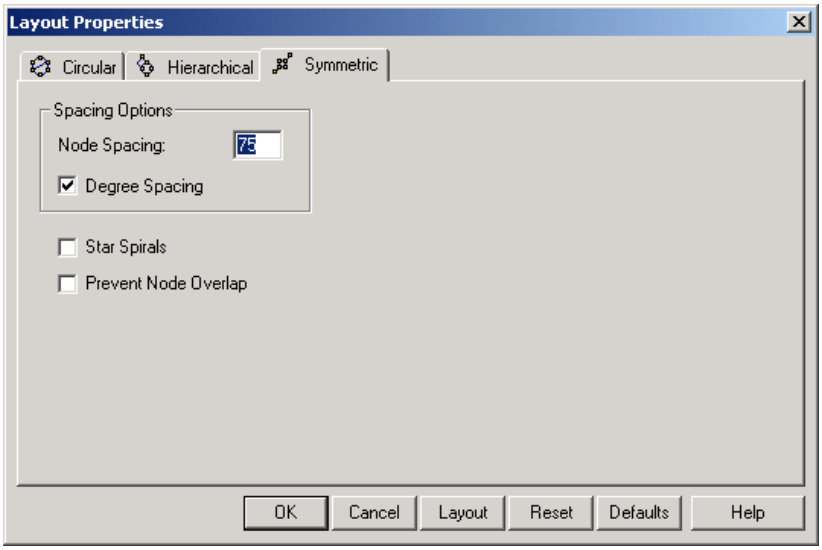


Figure 3.24 Settings for Symmetric layout

Field	Settings	Description
Spacing Options	Node Spacing	Provides a guide for displaying image density
	Degree Spacing	Reduces node crowding for highly connected nodes
Star Spirals	Checked = On Unchecked = Off	Puts nodes adjacent to a highly connected node in a spiral.
Prevent Node Overlap	Checked = On Unchecked = Off	Prevents nodes from overlapping

Table 3.3 Settings for Symmetric layout

Buttons

Table *3.4* describes the actions of the buttons in the Layout Properties dialog box.

Button	*Action*
OK	Saves any setting changes to the layout, applies it to the pathway in the Graphics window, and closes the Layout Properties box.
Cancel	Cancels any changes to the settings, returns the settings to previous, and closes the Layout Properties box
Help	Opens a help topic appropriate for dialog box options
Reset	Returns settings to previous
Layout	Applies the current settings to the pathway in the Graphics window
Defaults	Returns the settings to the default settings

Table 3.4 Button actions in the Layout Properties dialog box

Viewing Pathways in Text Format

In addition to displaying pathway elements graphically on the **Master View** tab, you can also display them in text format by clicking the **Text View** tab at the bottom of the Graphics window. The Text View tab provides a text summary of all the reactions, connectors, and components in a pathway as well as the annotations added to each. Information in the Text View tab is organized in hierarchical folders and, when you first click on this tab, the **Pathways** folder displays, with any objects selected in the Graphics window simultaneously selected in the Text View tab.

Note: When viewing a pathway in the Text View tab, the graphical tools in the Palette window and the Graphics toolbar are not available.

To view the contents in each folder, click the (**+**) sign to expand it. Click the (**-** sign to retract it.

Pathway folder—contains subfolders for the reactions and components that are included in the pathway. It also contains subfolders for each type of annotation that can be associated with a pathway including Organisms, Locations, and Cross Links (*Figure 3.25*). For more information about annotations, see *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37.

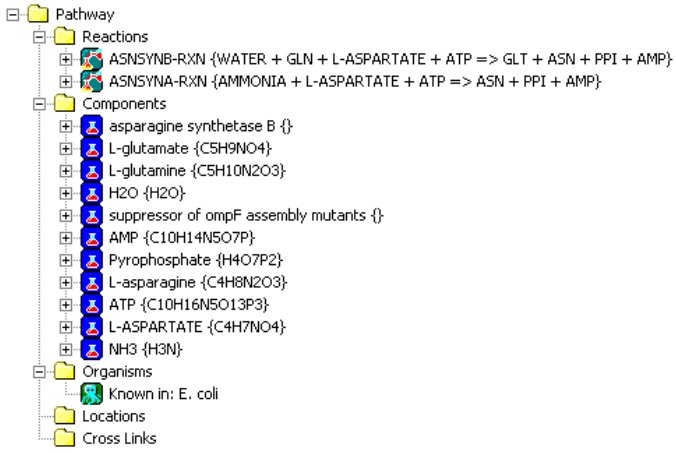


Figure 3.25 Pathway folder in the Text View tab

Reactions folder—contains subfolders for each separate reaction in the pathway (*Figure 3.26*). A reaction is represented by the icon. Each connector in the reaction is represented with its corresponding component by the icon. The Reactions folder also contains subfolders for each type of annotation that can be associated with a reaction including **Constants**, **Conditions**, **Locations**, **Organisms**, **Cross Links**, and **Pathways**. The properties of reactions and connectors can be modified from the Text View tab by selecting either a reaction or connector and selecting **Reaction Properties** or **Connector Properties** respectively from the shortcut menu.

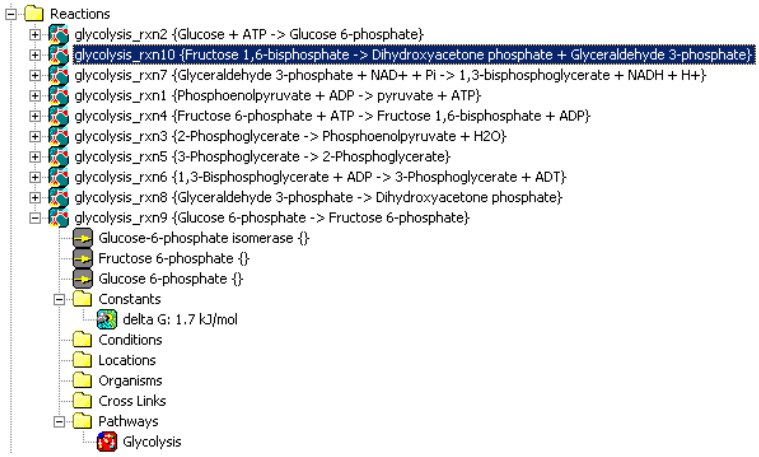

Reactions
glycolysis_rxn2 {Glucose + ATP -> Glucose 6-phosphate}
glycolysis_rxn10 {Fructose 1,6-bisphosphate -> Dihydroxyacetone phosphate + Glyceraldehyde 3-phosphate}
glycolysis_rxn7 {Glyceraldehyde 3-phosphate + NAD+ + Pi -> 1,3-bisphosphoglycerate + NADH + H+}
glycolysis_rxn1 {Phosphoenolpyruvate + ADP -> pyruvate + ATP}
glycolysis_rxn4 {Fructose 6-phosphate + ATP -> Fructose 1,6-bisphosphate + ADP}
glycolysis_rxn3 {2-Phosphoglycerate -> Phosphoenolpyruvate + H2O}
glycolysis_rxn5 {3-Phosphoglycerate -> 2-Phosphoglycerate}
glycolysis_rxn6 {1,3-Bisphosphoglycerate + ADP -> 3-Phosphoglycerate + ADT}
glycolysis_rxn8 {Glyceraldehyde 3-phosphate -> Dihydroxyacetone phosphate}
glycolysis_rxn9 {Glucose 6-phosphate -> Fructose 6-phosphate}
Glucose-6-phosphate isomerase {}
Fructose 6-phosphate {}
Glucose 6-phosphate {}
Constants
delta G: 1.7 kJ/mol
Conditions
Locations
Organisms
Cross Links
Pathways
Glycolysis

Figure 3.26 Reactions folder in the Text View tab

Components folder—contains subfolders for each separate component in the pathway. A component is represented by the icon and contains each reaction in which it is included (*Figure 3.27*). The properties of components can be modified from the Text View tab by selecting a component and then selecting **Component Properties** from the shortcut menu.

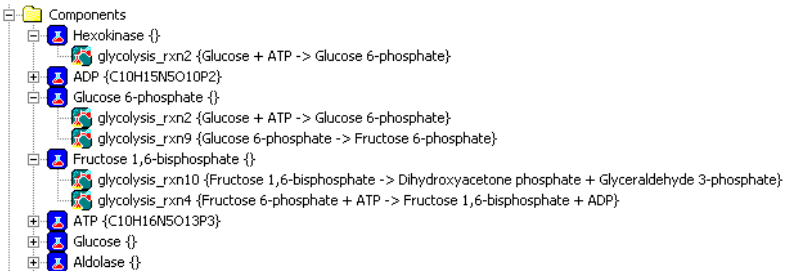

Components
Hexokinase {}
glycolysis_rxn2 {Glucose + ATP -> Glucose 6-phosphate}
ADP {C10H15N5O10P2}
Glucose 6-phosphate {}
glycolysis_rxn2 {Glucose + ATP -> Glucose 6-phosphate}
glycolysis_rxn9 {Glucose 6-phosphate -> Fructose 6-phosphate}
Fructose 1,6-bisphosphate {}
glycolysis_rxn10 {Fructose 1,6-bisphosphate -> Dihydroxyacetone phosphate + Glyceraldehyde 3-phosphate}
glycolysis_rxn4 {Fructose 6-phosphate + ATP -> Fructose 1,6-bisphosphate + ADP}
ATP {C10H16N5O13P3}
Glucose {}
Aldolase {}

Figure 3.27 Components folder in the Text View tab

The **Text View** tab contains no Experiments folders.

Creating Alternate Graphical Views

When you first open or create a pathway in the Graphics window, only the Master View and the Text View tabs display. You might build your pathway in the Master View and then decide that you want to use several different graphical versions of the pathway for different publication or teaching purposes. Vector PathBlazer allows you to create **Alternate Views** for these kinds of purposes that are stored within one pathway. An Alternate View can either be an exact copy of an existing view or a new view. When you copy an existing view, all of the graphical properties of that view are copied. When you create a new view, the default graphical properties of the Master View display. In an Alternate View, you cannot add components, connectors, or reactions to the pathway. You can, however, modify the graphical properties of the pathway elements and the graph itself, change the layout properties, hide pathway elements, and overlay gene expression

data sets on the pathway. Furthermore, if a component is added to a Master View after Alternate Views are created, the new component is "broadcast" or added to the Alternate Views. When you save the pathway and reopen it later, the Alternate Views are saved with the pathway and include the modifications you made in each.

Create a new Alternate View —from any tab, including the Text View tab or another Alternate View tab, by selecting **Tools > Manage Alternate Views > Create View**. Name the view in the dialog box that opens and click **OK**. A new tab is added to the Graphics window either next to the Text tab or next to the last Alternate View tab that was added. If you modified the display of the graphical elements in the Master View as in *Figure 3.28*, where common components are given similar shading and text styles, the graphical properties are removed in the new Alternate View and the default graphical properties display as in *Figure 3.29*. To save the new view to the pathway, click **Save**. For more information, see *Saving PathBlazer Components, Reactions and Pathways* on page 46.

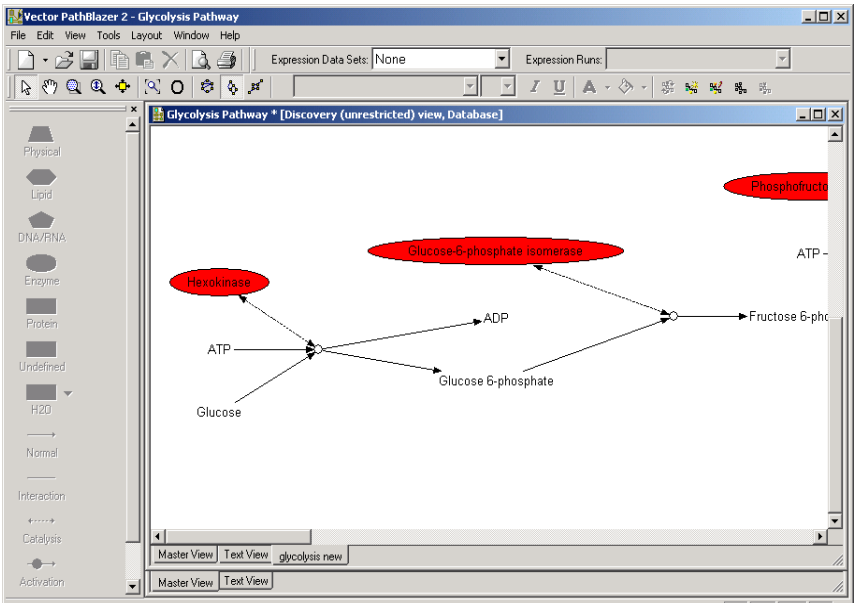


Figure 3.28 Modified graphical properties in the Master View

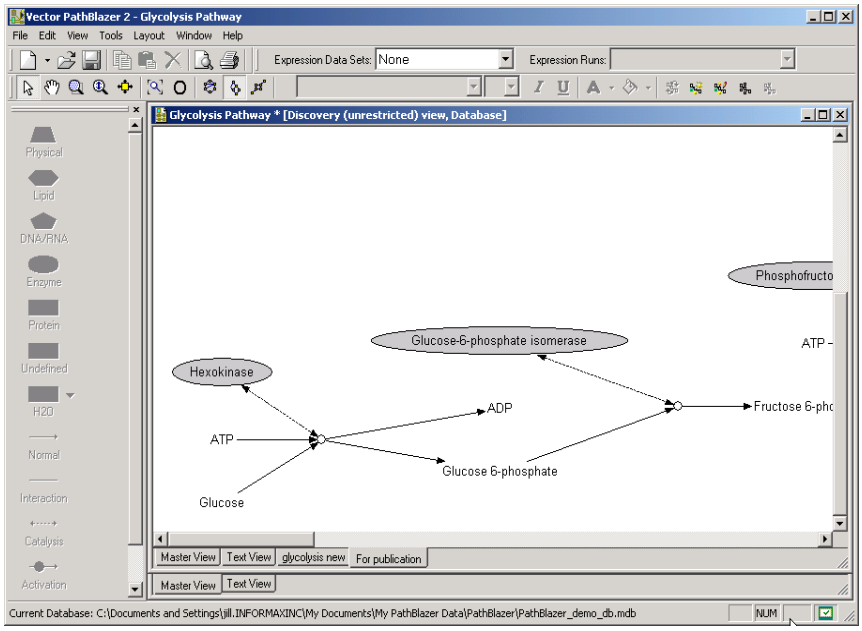


Figure 3.29 New tab is added and default graphical properties display in new Alternate View

***Copy a View*—**by selecting the graphical view, either the Master View or another Alternate View but not the Text View, you want to copy and selecting **Tool > Manage Alternate Views > Copy View**. Name the view in the dialog box that opens and click **OK**. A new tab is added to the Graphics window either next to the Text tab or next to the last Alternate View that was added and looks exactly like the view from which it was copied. To save the new view to the pathway, click **Save**.

***Delete an Alternate View*—**by selecting the tab of the view you want to delete and selecting **Tools > Manage Alternate Views > Delete View**. To save the pathway without the deleted view, click **Save**. The Master and Text View tabs cannot be deleted.

Working with Pathways in the Database Explorer

The Database Explorer has several main functions including browsing database contents, organizing data, and selecting data for display in the Graphics window.

Browsing Pathway Data

In Vector PathBlazer, there are two kinds of containers that you can use to organize data: folders and subsets. Each main data type (that is, Pathways, Reactions, Experiments and Components) in Vector PathBlazer displays in a folder in the List Pane (the left pane) of the Database Explorer. Each main folder contains a subset called the All Component/Reaction/Experiment/Pathway subset. A subset is a type of container that contains references to objects in the database and can be used to group objects with one or more properties in common. The All Component/Reaction/Experiment/Pathway subsets are system-defined subsets that reference each object of that type in the database. Any number of user-defined subsets can be created to organize objects.

***Browse data containers*—**by clicking the forward ( ) and backward ( ) arrow buttons on the Database Explorer toolbar. Move up a folder by clicking the folder button ( ). Folders and subsets display in the List Pane on the left of the Database Explorer and the objects they contain display in the Contents Pane on the right.

Objects in the Contents Pane either display as a list with details about each object or simply a list of objects. Click the **List** button ( ) on the Database Explorer toolbar to list the objects in the List Pane. Click the **Details** button ( ) to list properties of the objects in columns in the List Pane. The columns **Name**, **Description**, and **Formula** (for Components and Reactions only) display in the Contents Pane (*Figure 3.30*). Each object's name is listed in the **Name** column. If a description or formula has been entered, text also displays in these columns.

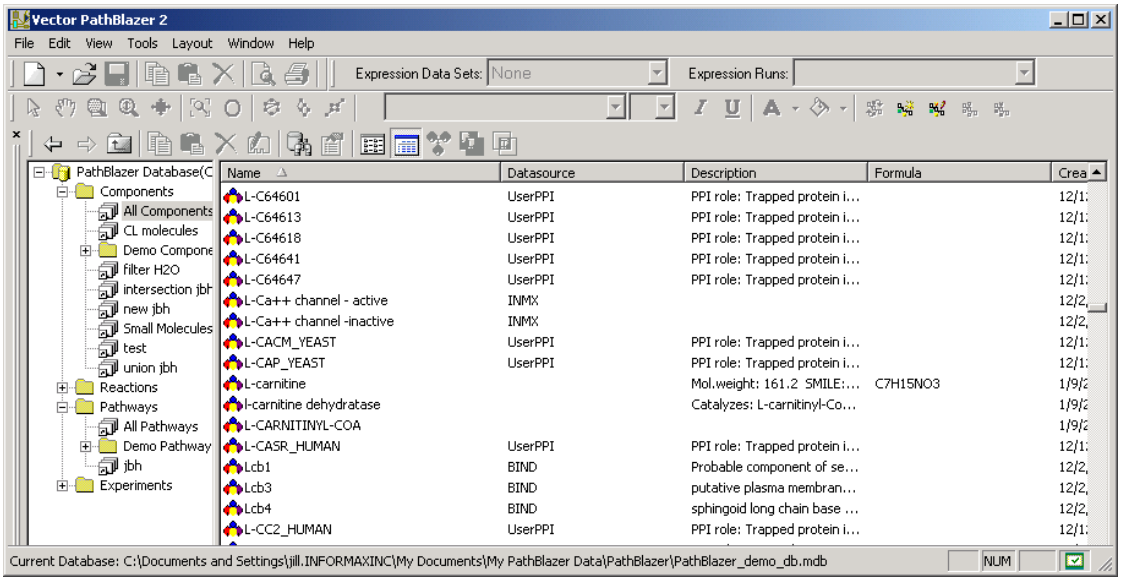


Figure 3.30 Database Explorer showing components in Details View; Original source database displays in the Datasource column

Sort a column—by clicking on the column header. An arrow is placed in the column header to indicate that sorting is based on that column. An up arrow designates an ascending sort order and a down arrow designates a descending sort order.

Resize column widths—by dragging the divider between columns to the left or right to reduce or enlarge the column width.

Remove columns from the display—by selecting **More** from the shortcut menu. In the **Column Settings** dialog box, a check next to the column name means that it is displayed (*Figure 3.31*). To hide columns from the display, uncheck the box next to the column name. The Name column cannot be hidden. You can also select a column name and click **Hide** to hide it or **Show** to display it.

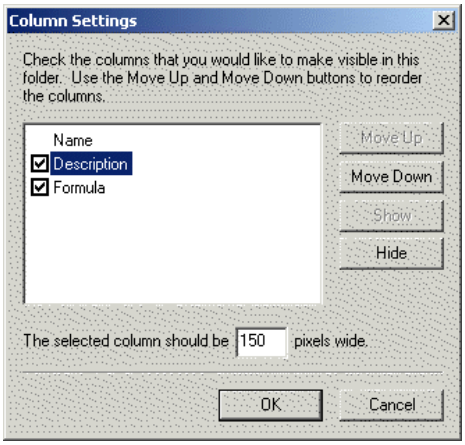


Figure 3.31 Column settings dialog box for customizing column display in Database Explorer

Rearrange columns—by selecting a column name in the Column Settings box and clicking **Move Down** or **Move Up**. In the List Pane, you can also drag the Description or Formula column headers left or right. The Name column is fixed and cannot be reordered.

Naming, Copying, and Deleting Objects

In Vector PathBlazer, the unique identifier of an object is its name or its “primary” name and there can be only one object in the database with a particular primary name. Components can have synonyms, or alternative names, as “secondary names”. While a primary name can only be associated with one object, a synonym can be associated with more than one object. For example, you might want to enter the stereoisomers of a sugar such as D- and L-glucose as separate components in the database. Then you might assign the synonyms ‘glucose’ and ‘mannose’ to them. As you will learn more about in Chapter 5 *Drawing Pathways*, you can add a component that already exists in the database to the Graphics window and Vector PathBlazer will search the database by primary name and by synonym to retrieve the component from the database. All components that match by name or synonym will be listed in the search. However, the primary name and not the synonym displays in the Graphics window when components are drawn and when they are listed in the Database Explorer.

When renaming, copying, or deleting an object from the Database Explorer, the following rules apply:

- The All Components/Reactions/Experiments/Pathways subsets contain the original or “primary” copy of any database object. When a copy is made of one of these objects and placed in a subset, the subset contains a reference or shortcut to the primary object and the copied reference is named identically to the referenced object.
- When an object in the database is renamed or its properties are changed, the primary object and all references to that object are also changed regardless of whether the object is changed from the primary copy or a referenced copy. If a component is referenced by a reaction, the name of the component is changed in the reaction. If a reaction or component is referenced by a pathway, the name of the component or reaction is changed in the pathway.
- When an object is deleted, if it is a reference to the primary object (that is, if it is not deleted from the All Components/Reactions/Experiments/Pathways subsets) then the reference is deleted but the primary object is not. A primary object can only be deleted if it is not referenced by any copies. To permanently delete an object, it must be deleted from the All Components/Reactions/Experiments/Pathways subsets; deleting it from a subset only deletes the reference. If a component is included in one or more reactions or pathways, it cannot be deleted until it is removed from all reactions and pathways in which it is included. If a reaction is included in one or more pathways, it cannot be deleted until it is removed from all pathways in which it is included. Pathways can be deleted without affecting associated components or reactions.

Rename an object—by double-clicking on the name. Enter a new name at the cursor.

Copy an object—by selecting it and then clicking the **Copy** button ( ) in the Explorer toolbar or selecting **Copy** from the shortcut menu. Paste the reference to the object into another subset by clicking the **Paste** button ( ) or selecting **Paste** from the shortcut menu. You can also select objects in the List Pane and drag and drop them into a subset in the Contents Pane.

Delete an object—by selecting it and then clicking the **Delete** button ( ) on the Explorer toolbar or selecting **Delete** from the shortcut menu. Click **Yes** in the confirmation dialog box that opens.

Organizing Pathway Data

Subsets and folders can be used to organize the data contained in the database. The Components/Reactions/Pathways folders and the All Components/Reactions/Experiments/Pathways subsets are system-defined containers and cannot be renamed or deleted. However, any number of user-defined folders and subsets can be created.

Creating Folders

Folders are intended to organize subsets and subfolders. A folder can only be created in another folder; it cannot be created in a subset.

Create a folder—by selecting a folder in the Contents or List Pane (select the Components/Reactions/Pathways folder if there are no other folders) and then selecting **Create Folder** from the shortcut menu. Name the folder and press ENTER.

Delete a folder—by selecting it in the Contents or List Pane and then selecting the **Delete** button ( ) on the Explorer toolbar or **Delete** from the shortcut menu. Click **Yes** in the confirmation dialog box.

Creating Subsets

Subsets can only contain one object type. For example, a subset created in the Components folder can only contain components. When a subset is selected in the Explorer Contents Pane, the number of objects in the subset (and displayed in the List Pane) displays on the status bar. Subsets are contained in folders and cannot be contained in another subset. When creating a subset from the List Pane, you can either create an empty subset or you can select objects and add them to a new subset. You can also create a subset based on search results. For more information about searching the database, see *Searching Objects in the Database and Creating Subsets* on page 54.

Note: If you try to assign a name already given to an existing subset, you will be informed that you must specify a different name.

Create an empty subset—by selecting a folder in the Contents or List Pane and then selecting **Create Subset** from the shortcut menu. A subset initially called New Subset is added to the List Pane. Name the subset and press ENTER.

Create a subset with specific contents—by selecting one or more objects in the List Pane and then selecting **Create Subset** from the shortcut menu. Select a list of consecutive objects by selecting the first object, pressing the SHFT-key, and selecting the last object. Select non-consecutive objects by pressing the CTRL-key and selecting the objects. In the **Create Subset** dialog box, select the folder to contain the new subset, enter a name and description, and click **Create** (*Figure 3.32*). The subset is created in the List Pane and contains the selected objects.

You can create a new subset containing all of the items in two or more existing subsets of like object types (union) or the items common to two or more existing subsets of like object types (intersection).

- To create a new subset that contains *all the items of two or more existing subsets*, highlight at least one subset in the List Pane, then click the **Union** button ( ) on the Explorer toolbar. Check the two subsets whose contents you want to combine. Click the **Results Subset** button, and in the dialog box that opens, enter the **Name** of the new repository subset. Click **OK**.
- To create a new subset containing *items common to two or more existing groups*, highlight at least one subset in the List Pane, then click the Intersection button ( ) on the Explorer toolbar. Check the two subsets whose common contents you want to combine in the intersected subset. Click the **Results Subset** button, and in the dialog box that opens, enter the **Name** and **Description** of the new repository subset. Click **OK**.

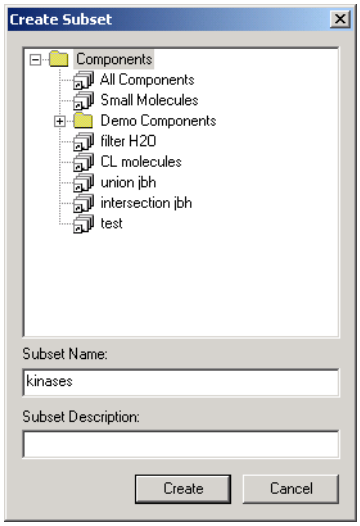


Figure 3.32 Create subset dialog box

Delete a subset—by selecting it in the List or Contents Pane and clicking the **Delete** button ( ) on the Explorer toolbar or **Delete Subset** from the shortcut menu. Click **Yes** in the confirmation dialog box that opens. The All Components/Reactions/Experiments/Pathways subsets cannot be deleted.

In addition to creating subsets of specific components, reactions, experiments or pathways, you can also create subsets of all reactions in a pathway, all components in a pathway, or all components in a reaction.

Create a subset of all reactions in a pathway—by selecting one or more pathways in the List Pane and then selecting **Create Reaction Subset** from the shortcut menu. In the Create Reaction Subset dialog box that opens, enter a name and description for the reaction subset and click **Create**. All reactions included in the selected pathways are added to the new subset.

Create a subset of all components in a reaction or in a pathway—by selecting one or more reactions or pathways (reaction and pathways cannot be displayed at the same time) in the List Pane and then selecting **Create Component Subset** from the shortcut menu. In the Create Component Subset dialog box that opens, enter a name and description for the component subset and click **Create**. All components included in the selected reactions or pathways are added to the new subset.

Reversing the Direction of a Reaction

In Vector PathBlazer, two reactions are required to represent a reversible reaction: one reaction in which a group of components are substrates and another group are products and a second reaction in which the substrates and products are switched. In some cases, you may want to reverse the direction of a reaction (that is, make the substrates products and vice versa) without rebuilding the reaction from scratch. If you want to swap the substrates for products in an imported reaction, you can easily reverse the reaction in Vector PathBlazer.

When the direction of a reaction is reversed, a new reaction is created in the database. Use the following steps to reverse the direction of a reaction.

1. Select the reaction in the List Pane of the Database Explorer. To display it in the Graphics window, double-click on it or select **Open** from the shortcut menu. In the following example, the reaction in *Figure 3.33* is reversed.

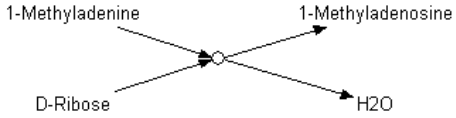


Figure 3.33 Reaction as it displays in Graphics window before it is reversed

Note: Protein-protein interactions cannot be reversed.

2. Select **Reverse Reaction** from the shortcut menu. The Reaction Properties dialog box opens to create a new reaction (*Figure 3.34*). In the **Name** field, the characters '_R' are automatically appended to the name of the reaction to indicate it is a reverse reaction. If a formula is entered in the original reaction, the **Formula** field displays the reverse of the original reaction. Any other annotations that were associated with the original reaction remain with the reverse reaction. Add or modify any annotation, including the name, and then click **OK**.

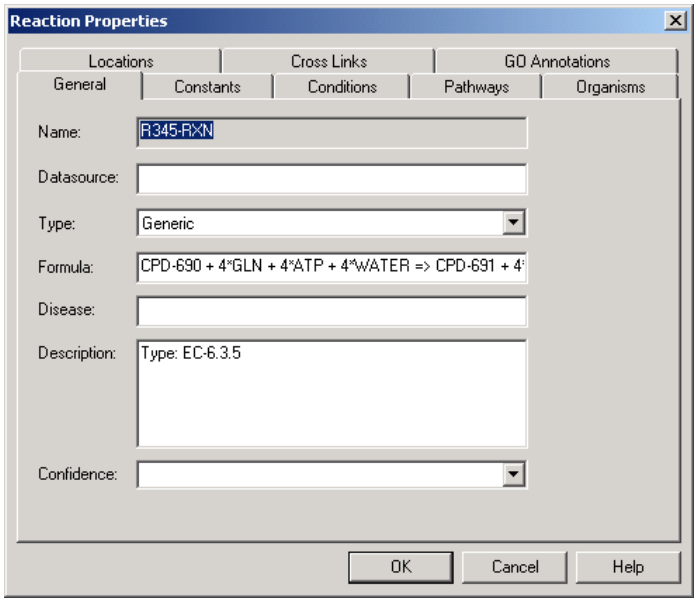


Figure 3.34 Reaction Properties dialog box

The reverse reaction is added to the All Reactions subset.

Adding Pathways, Reactions, Experiments, and Components to the Graphics Window

The Database Explorer window interacts with the Graphics window by allowing you to select objects in the List Pane and either drag them or open them so that they are drawn in the Graphics window. The following methods are described in more detail in the context of drawing pathways in Chapter 5 *Drawing Pathways*, but they are briefly listed here.

- One component at a time can be dragged from the List Pane and dropped into the Graphics window. The Graphics window can either be blank (that is, a new Graphics window) or can display one or more components, reactions, pathways or experiments. (Experiments display only when a pathway is open.) You can drag and drop more than one component into a single Graphics window but you cannot select multiple components in the List Pane and drag them all into the window at once. Use the instructions in *Adding a Reaction* on page 122 to connect a component to another component in a reaction or pathway.

Note: If the component or reaction you are adding to a pathway or reaction is already present in the displayed pathway, an Add Reaction dialog box opens displaying the duplicate object and allows you to resolve the issue in either of two ways:

- Check the **Pool** checkbox to link the object with the existing reaction or pathway.
- Check the **Do not Pool** checkbox to maintain the duplicate object in a separate reaction or pathway in the Graphics Pane.

- Reactions and pathways can be opened in the Graphics window from the List Pane by either double-clicking on a reaction or a pathway or selecting **Open** from the shortcut menu. Each subsequent reaction or pathway that is opened from the Database Explorer is opened in a new Graphics window. To add reactions to a displayed reaction, see *Adding a Reaction* on page 122. To add components or reactions to a pathway, see *Adding a Component* on page 113.

Annotating Pathways, Components, Experiments, Reactions, and Connectors

An annotation in Vector PathBlazer is a property or an attribute that can be added to an object. Annotations can be useful for recording pertinent information about an object or for searching for objects in the database that all have a property in common. For example, a search of Epidermal Growth Factor (EGF) on OMIM (http://www.ncbi.nlm.nih.gov/entrez/dispomim.cgi?id=131530) displays a summary of what is currently known about EGF: it has a role in growth control and has been implicated in malignant melanoma. In addition to the interactions EGF makes with other known proteins (namely the EGF receptor which mediates a cascade of signal transduction events) that can be stored in the Vector PathBlazer database, each of these known properties can be included as annotations to EGF. When objects are imported into the database, many annotations are automatically imported with the objects. For more information about importing data, see *Chapter 4*.

Each object type in Vector PathBlazer, including pathways, components, experiments, reactions, and connectors, has a specific set of fields that can be associated with a particular object type. Some fields have a pre-defined set of values and other fields accept other formats such as text strings and numbers.

Annotations can be added to an object during several different operations in the program including saving and viewing objects:

In the Properties dialog box—most tabs have an **Add** button. Click the **Add** button and in the dialog box that opens, make the appropriate selections (values display in the following table, Table *3.5*.)

While saving an object—when an object is saved (by selecting **File > Save As** when a pathway or reaction is open in the Graphics Window) a wizard guides you through adding annotations by field. *Figure 3.35* shows the Save dialog box for pathways and reactions. The hyperlinks listed in the left pane each correspond to a different type of annotation. The available fields in each annotation screen depend on the type of annotation. The **Next** and **Back** buttons advance the annotation screens in order and the hyperlinks in the left pane jump to the corresponding screen.

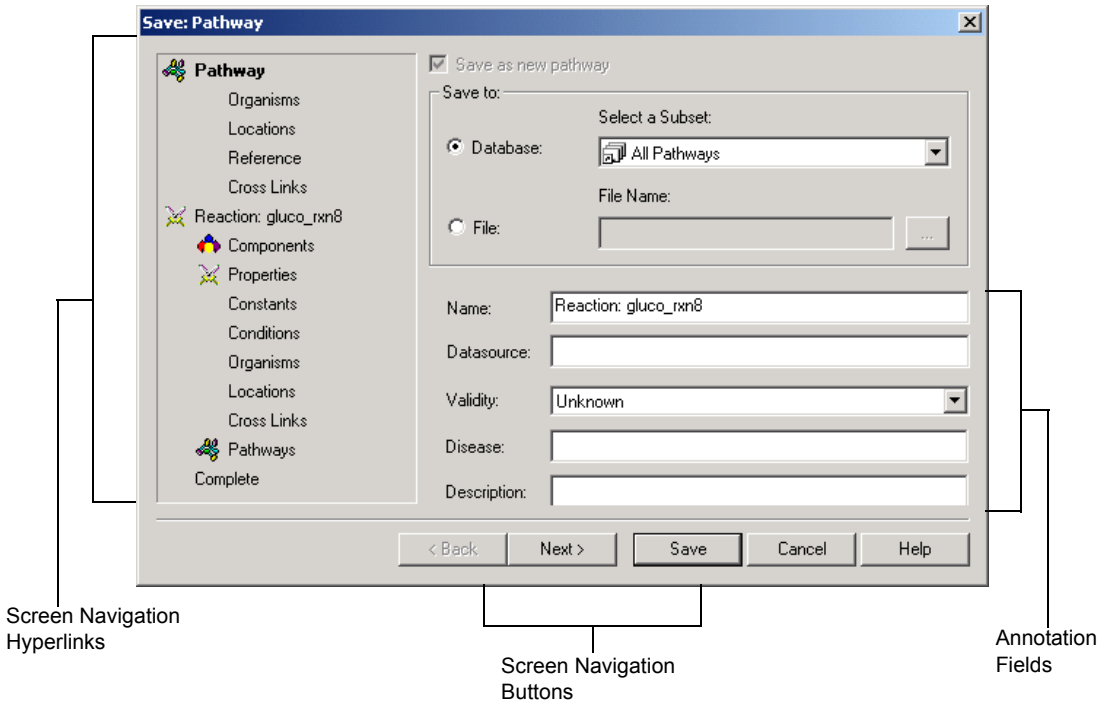


Figure 3.35 Annotations are listed in screens when saving an object

When an object has been saved to the database, its properties are viewed by selecting **View > Properties**, **Properties** from the shortcut menu in Database Explorer, or **Component/Connector/Reaction/Experiment/Pathway Properties** in the Graphics window, **Master View**. A screen containing tabs for each attribute type lists the attributes by field. *Figure 3.36* shows the Properties box for a pathway.

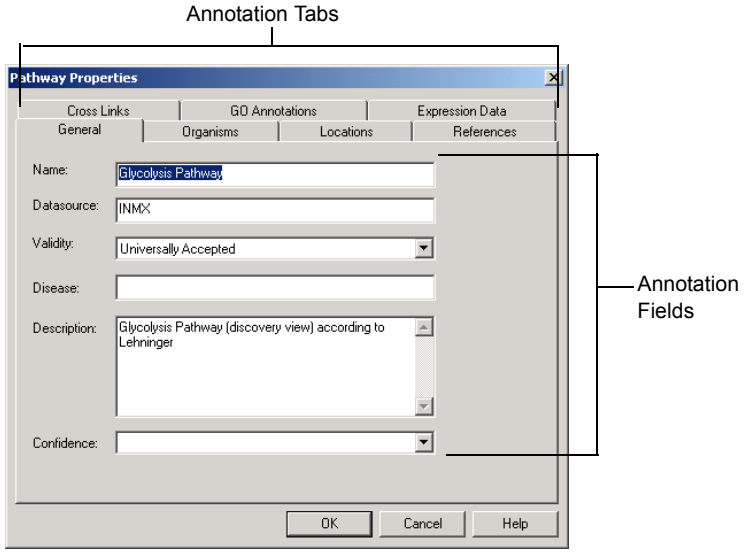


Figure 3.36 Annotations are listed in tabs when viewing the properties of a saved object

In PathBlazer Database Explorer—to batch-change annotations to contents of subsets of either components or reactions, in the Details View of PathBlazer Explorer (left pane), open the appropriate **Component** or **Reactions** folder by clicking the (+) to its left. Right click the component or reactions subset and choose **Content Properties**. The dialog box that opens displays attributes that apply to all contents of that subset (*Figure 3.37*).

On any of the tabs, **Organism**, **Location**, or **Description** review, change or add any of the attributes. On each tab, select the appropriate radio button: **Don't Apply**, **Append**, or **Replace** [an existing annotation]. Note that any changes you make will be assigned to ALL of the selected subset(s)' contents. Click **OK** to apply the changes.

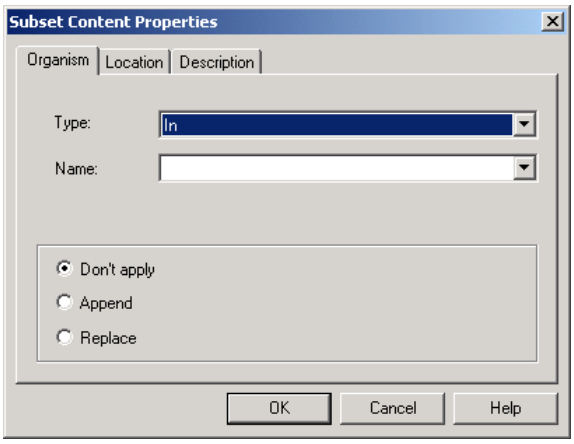


Figure 3.37 Subset Content Properties dialog for reviewing or applying batch annotations

Annotation Fields for Components, Reactions, and Pathways

Many of the annotation fields for pathways, components, and reactions are the same. For example, each of these objects can have a location associated with them. Table *3.5* lists the annotation fields, values, and descriptions for pathways, components, and reactions first by the screen or tab in which they are found. Annotation fields for connectors are listed later in this section.

Tab/ Screen	***Field***	***Description/ Value(s)***	***Pathway=P Reaction=R Component=C***
General/ Component	Name	Primary name of the object Example:Glucose String	PRC
	Datasource	Origin of the object. Example: KEGG String Note: If an entry is imported from two data sources (for example, KEGG and DIP), the data source displays both sources. For example, if a component was first imported from KEGG and then from DIP, this field displays "KEGG, DIP".	PRC
	Description	String	PRC
	Disease	Disease or condition associated with the object String	PRC
	Chemical Formula	Chemical formula of a component Example: C10H15N5O10P2 (ADP) String	C

Table 3.5 Annotation fields and values for pathways, components, and reactions

Tab/ Screen	***Field***	***Description/ Value(s)***	***Pathway=P Reaction=R Component=C***
General/ Component (cont'd)	Source	Derivation of a component Values: Biological Synthetic	C
	Formula	Formula of a reaction Example: Chloroacetic acid + H2O <=> HCl + Glycolate String	R
	Type	Type of a reaction Values: Generic — Signal Transduction Metabolic — Unknown	R
	Confidence	Description: level of confidence in this reaction Values: Theoretical (guess) Unlikely — Probable Possible — Universally accepted	R
	Validity	Significance of a pathway Values: Unknown — Hypothetical Doubtful — Novel Experimental Test — Universally accepted	P
Organisms	Type	Designates how definitively it is known if an object is present in an organism Values: **In**: Definitively known to be in one or more organisms. If an object is in one or more organisms, all others are excluded. **Known in**: Known to be in an organism but all others cannot be ruled out. **Not in**: Opposite of Known in. Known not to be in an organism but all others cannot be ruled out.	PRC

Table 3.5 Annotation fields and values for pathways, components, and reactions (Continued)

Tab/ Screen	***Field***	***Description/ Value(s)***	***Pathway=P Reaction=R Component=C***
Organisms (cont'd)	Name	Species name Values: *Arabidopsis thaliana* *Bos taurus* *Caenorhabditis elegans* *Homo sapiens* *Rattus norvegicus* *Saccharomyces cerevisiae* *Mus musculus* *Schizosaccharomyces pombe* *Danio rerio* *Takifugu rubripes* *Dictyostelium discoideum* *Neurospora crassa* *Xenopus laevis* *Drosophila melanogaster* *Zea mays* *Escherichia coli* *Plasmodium falciparum* *Oryza sativa*	PRC
Cross Links	Display Name	Name that displays on the shortcut menu in the Graphics window for a selected object. If no name is entered, the Accession ID or the URL displays. See fields for Accession ID and URL below. String	PRC
	Type	Specifies a link from an object in the Vector PathBlazer database to either the Vector NTI database or to a URL Values: Database URL	PRC
	Database (Type = Database)	Opens object with the corresponding object name in the Vector NTI Suite or Advance database (if installed). See also description of Accession ID field. <u>Values:</u> <u>Component</u> VNTI (DNA/RNA) VNTI (PROTEIN) VNTI (CITATION) VNTI (BLAST) <u>Pathway and Reaction</u> VNTI (CITATION)	PRC
	Accession ID (Type = Database)	Unique object name in the VNTI Suite/Advance database. Only names of DNA/RNA and protein molecules, citations, and Blast results can be linked from Vector PathBlazer to the VNTI database. Example: GAL4_YEAST String	PRC

Table 3.5 Annotation fields and values for pathways, components, and reactions (Continued)

Tab/ Screen	***Field***	***Description/ Value(s)***	***Pathway=P Reaction=R Component=C***
Crosslinks (cont'd)	URL (Type = URL)	Fully qualified URL. Example: http://www.expasy.org/cgi-bin/get-enzyme-entry?5.4.99.3 String **Note**: When KEGG, BIND, TransPath, BioCyc and DIP entries are imported, one or more URLs is automatically created for entries in these databases. For more information, see *Pre-Defined URLs* on page 107	PRC
Locations	Type	Designates how definitively it is known if an object is present in a location Values: Known in Not in In See definitions in Organism field.	PRC
	Tissue	Designates which tissue an object is known to occur in String	PRC
	Organelle	Designates which organelles an object is known to be in Values: cell-wall centriole centrosome chloroplast chromatin cilia cis-golgi cytoplasm cytoplasmic-membrane cytoskeleton endosome ER-general ER-rough ER-smooth extracellular flagella Golgi golgi-stack lysosome medial-golgi mitochondrion nuclear-pore nucleolus nucleus nucleus-inner-membrane nucleus-outer-membrane outer-membrane peroxisome plastid ribosome trans-golgi vacuole vesicle	PRC

Table 3.5 Annotation fields and values for pathways, components, and reactions (Continued)

Tab/ Screen	***Field***	***Description/ Value(s)***	***Pathway=P Reaction=R Component=C***
GO Annotations	Source Database	Original source of the term or the annotation Example: Term: http://www.godatabase.org/dev/database/archive/latest/	PRC
	Unique ID in Database	ID in the original database	PRC
	Evidence Type	Hierarchy of evidence or confidence in the validity of the annotation	PRC
	Organism	Source organisms for the annotation. Organisms are listed from top to bottom in the order of most frequently used.	PRC
Component Class	Component Class	Designates the type of molecule Values: Physical Protein Protein Enzyme DNA RNA Small inorganic molecule/ion Small organic molecule/ion Unknown	C
	Protein Subclass: (Protein only)	Designates the type of protein Values: Enzyme Regulatory Structural Unknown	C
	RNA Subclass (RNA only)	Designates the type of RNA Values: Unknown mRNA tRNA rRNA	C
	E.C. Number: (Protein only)	Enzyme Commission String	C
	Generic Name (Protein only)	String	C
Synonyms	Synonym	Alternative names of an object String	C
References	None	Any comment about an object String	P

Table 3.5 Annotation fields and values for pathways, components, and reactions (Continued)

Tab/ Screen	*Field*	*Description/ Value(s)*		*Pathway=P Reaction=R Component=C*
Constants	Name	Constants that can be associated with reactions Values: K_a (association) K_b (complex formation - reverse)	 K_d (dissociation) K_{eq} (equilibrium) K_f (complex formation - forward) K_m (Michaelis) v_{max} (max velocity)	R
	Value	Value of the constant Number		R
Condition	Name	Condition that can be associated with reactions Values: pH range Temperature range		R
	Value	Value of the condition Number		R
Pathway	Pathway Name	Pathways that are associated with reactions String		R
Expression Data	Expression database file	Reference to original database, if available		P

Table 3.5 Annotation fields and values for pathways, components, and reactions (Continued)

Annotation Fields for Connectors

The annotations for describing connectors are more limited than those for pathways, reactions and components and are all contained in one dialog box (*Figure 3.38*). Change the annotations of a connector by selecting it in the Graphics window and then selecting **Connector Properties** from the shortcut menu to open the Connector Properties box.

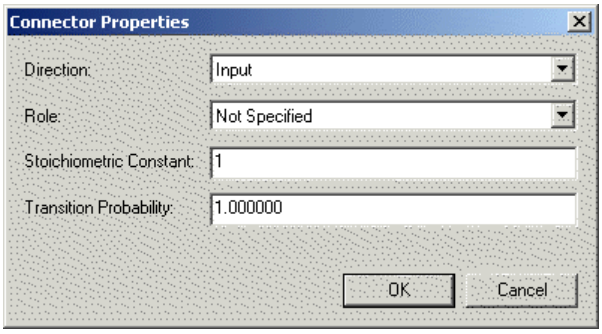


Figure 3.38 Connector Properties dialog box

Table *3.6* lists the annotation fields and values for connectors.

Field	*Description*
Direction	Designates the direction of the connector Values: Not Specified Input Output Input/Output
Role	Designates the role of the connector. Only valid if Direction = Input. Values: Not Specified Normal Activating Inhibiting
Stoichiometric Constant	Quantity of a component participating in the reaction Number
Transition Probability	Can be used to describe the probability of a change of state Number

Table 3.6 Annotation fields and values for connectors

Merging Components Manually

Components and reactions are automatically merged during data import. This automatic merge is not infallible, however. Source databases utilize different data models, different substance classifications, etc., and it is inevitable that some components which should be merged will not be, while others will be merged incorrectly.

All merge events are recorded in a log file, described on page 71. If two components are merged incorrectly during import, you can manually re-create a missing component and link it to appropriate reactions. There is no automatic way to 'un-merge' two components.

After data has been imported into Vector PathBlazer, you can manually merge components using a Merge ‘Wizard’. Use the following steps to manually merge components.

1. Select one component in the List Pane of Database Explorer, and select **Merge Components** from its associated shortcut menu.
2. In the Merge Components dialog box that opens, **Component 1** displays the component you selected. For **Component 2**, browse and locate a component in a subset. Click **Next** to continue.
3. In the second screen of the Merging Components dialog box, unique attributes of both components are listed. Select attributes that are to be included with the final merge product. Attributes that can be selected are: **Name**, **Chemical Formula**, **Source**, and **Component Class**. Click **Next** to continue.
4. The next dialog box lists non-unique attributes as text strings which you can edit. Click **Next** to continue.
5. The next several dialog boxes display annotations for each component (with each dialog box assuming the name of the annotation): **Component Locations**, **Organisms**, **Compo-**

nent Crosslinks, **Synonyms**, and **GO Annotations**. You can edit any of these annotations. Continue to click **Next** to continue to each succeeding dialog box.

6. Before you complete the merge, a Merging Components dialog box describes the merge that will occur, which component will be deleted from the database and which will be retained as well as how components have been renamed, where appropriate. Click **Finish** to execute the merge.

If there are conflicts in the organism or location attribute, an error message describing the conflict displays when you try to continue.

Note: Components and reactions can also be merged automatically during import. For more information, see *Merge Option Dialog Box* on page 67.

Saving PathBlazer Components, Reactions and Pathways

When you create or modify a component in the Graphics window, you are automatically prompted to name, annotate, and save the component. If you decide not to save the pathway in which a component is drawn, the component is still saved to the database. Details for saving new components are described in *Drawing a New Component* on page 113. Pathways and reactions are saved differently than components. New or modified pathways and reactions are saved using the **Save** command. A wizard, similar to that used for saving components, is used to save pathways and reactions. Pathways and reactions are saved as independent objects but are saved using the same wizard.

Saving a Pathway or Reaction to the Database or a File

To save a pathway, all components must be connected to at least one reaction. However, separate reactions in a pathway do not have to be connected to another reaction or do not have to be saved as part of a pathway. Use the following steps save pathways and reactions.

1. Select **File > Save** or click the **Save** ( ) button on the toolbar. If the pathway has already been saved, any changes will overwrite the existing pathway.

   To save a pathway that has not been previously saved or to save the pathway under a different name, select **File > Save as**. The **Save** dialog box opens and includes a number of options (*Figure 3.39*).

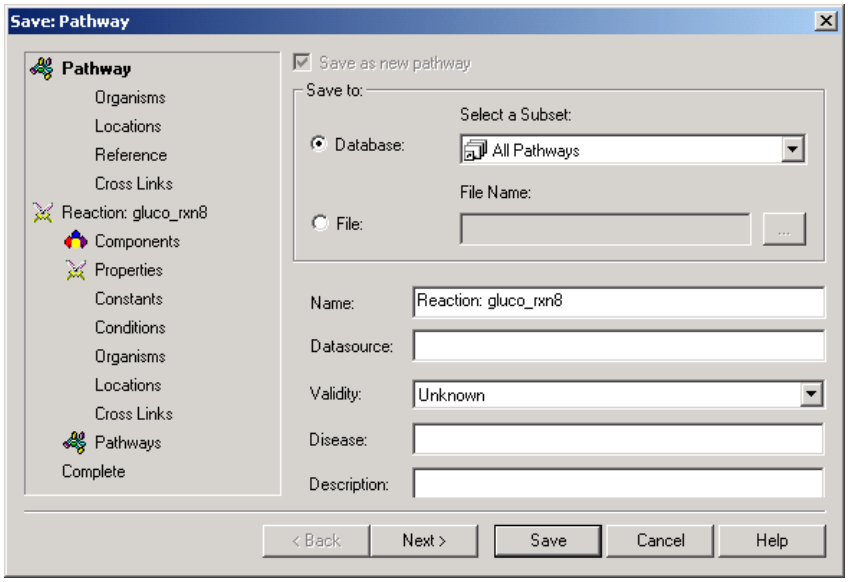


Figure 3.39 Dialog box for saving pathways and reactions

The Save dialog box can be used to save pathway and reactions to either the database or to a .pw file and can also be used to annotate pathways and reactions, similar to annotating components. A .pw file is an XML file in which individual pathways are saved to the local file system. These files can be used to archive and share pathways with other Vector PathBlazer users.

The Save dialog box is divided into two parts: a wizard displays on the right side and contains **Back** and **Next** buttons for advancing through the screens in sequence. The left side contains hyperlinks for jumping to specific screens. The left side is also divided into two parts. The hyperlinks under **Pathway** are for naming and annotating the current pathway. Each reaction in the pathway displays further down under **Reaction: <Reaction Name>**. The hyperlinks under a reaction name are for naming and annotating that reaction. A separate screen displays with the appropriate annotation for each hyperlink.

2. To save the pathway to the database, select the **Database** radio button and select a specific subset in the drop-down menu. If you do not select a subset, the pathway is saved to the **All Pathways** subset.

   To save a pathway to a file, select the **File** radio button and click the **Browse** button (...). In the **Save as** dialog box that opens, navigate to the location where you want to save the file and enter a name in the **File name** field. All files are saved with a .pw extension to indicate they are Vector PathBlazer files, which means they can be shared or reopened in the program. Click **Save**. The complete path to the file displays in the **File Name** field.

3. To save the pathway and its reactions without annotating them or changing any existing annotations, click **Save**. The pathway and reactions are saved to the database or specified file. Note that the pathway is saved as the default name and the reactions are unnamed. To name the reactions and annotate the pathway, reactions, and components, continue to the next step.

Important: If reactions in a pathway are not named, they CANNOT be saved as independent objects in the database. Use the remaining steps to name any unnamed reactions. You can, however, save reactions not going through a pathway. See *Saving Reactions Not Going Through a Pathway* on page 50.

4. In the first screen, the **Save:Pathway** screen, enter information in the **Name**, **Database**, **Validity**, **Disease**, and **Description** fields (*Figure 3.39*). The **Name** field is required. The **Pathway** hyperlink is highlighted in the left pane. Click **Next** to move to the next screen in the sequence, which is the **Organisms** screen, or click any hypertext link in the left pane to go directly to that screen.

5. The Organisms, Locations, Reference, and Cross Links screens work in the same way and each is highlighted in the left pane when it is the displayed screen. The Organisms screen is shown in *Figure 3.40* to illustrate how to add an annotation using one of these four screens. Each annotation field is described in *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37.

In the **Save:Pathway Organisms** screen, associate one or more organisms with the pathway by clicking **Add**.

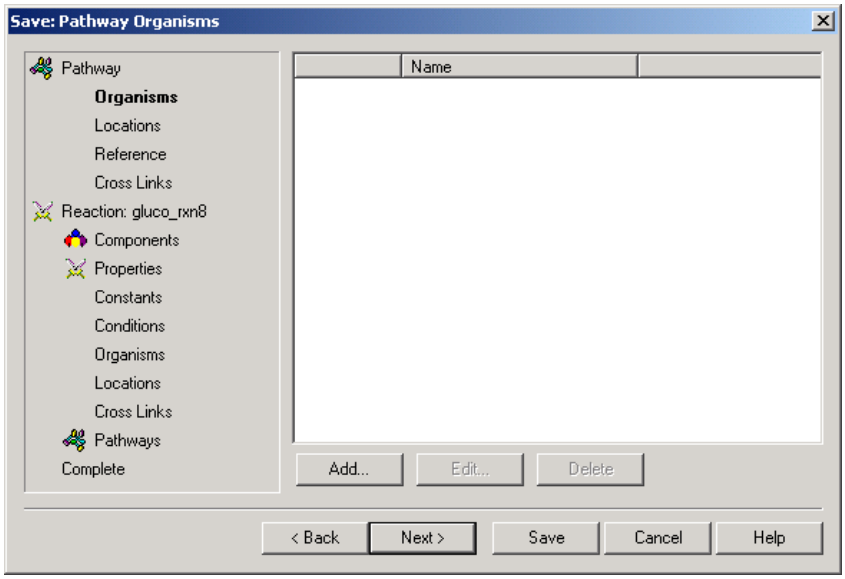


Figure 3.40 Associating a pathway with one or more organisms

In the **Organism** dialog box, select a **Type** and a **Name** from the drop-down lists (*Figure 3.41*). You can also type in the name of an organism if it is not listed in the drop-down list. Click **OK**. Select an associated organism and click **Edit** to modify it. Click **Add** to add another organism. Click **OK**. Click **Next** or click a hyperlink in the left pane to move to the next Pathways annotation screen. Continue annotating the pathway from the remaining screens.

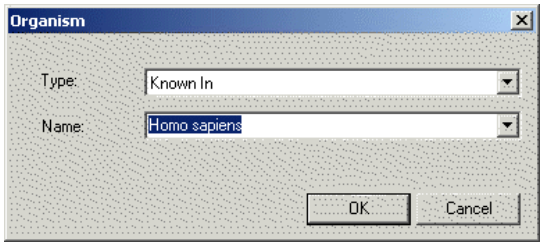


Figure 3.41 Assigning organism attributes to a pathway

6. If you advance the Pathway screens in sequence, the Reactions screens display next. For each reaction, the **Save:Reaction Components** screen displays first and contains a list of the components in the reaction (*Figure 3.42*). The Name column displays each component and the remaining columns display information about the connector to which the component is associated in the reaction.

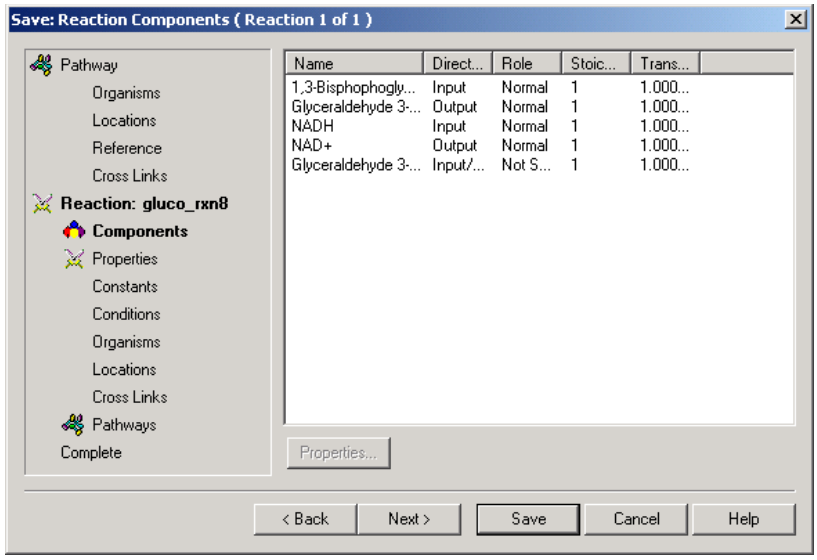


Figure 3.42 Components screen in the Save dialog box that lists all components in a reaction

To view the properties of any connector in a reaction, select the component and click **Properties**. The Reaction Component dialog box lists the properties of a particular connector in a reaction and the component to which it is connected (*Figure 3.43*). To change the component, click the **Browse** button (...) and create a new component or select one from the database. To change any of the properties of the connector, change the values in the remaining fields. Click **OK**. Click **Next** to move to the next screen or click one of the hyperlinks to jump to a screen.

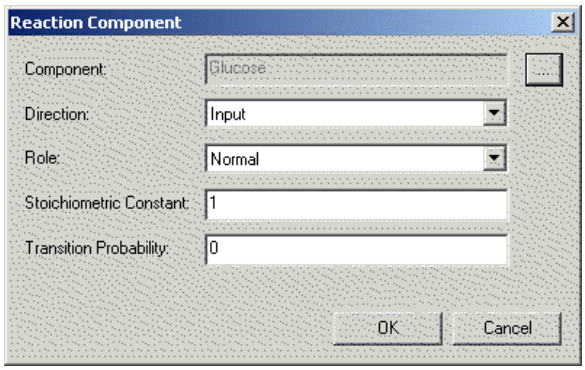


Figure 3.43 Reaction components dialog box

7. The Properties screen for the reaction displays. This screen is similar to the first screen for annotating a pathway. Enter information in the fields. The **Name** field is required. Click **Next** or click a hyperlink to move to the next screen.

   The method of entering information in the remaining screens, **Constants**, **Conditions**, **Organisms**, **Locations**, **Cross Links**, and **Pathways**, is the same as the method described in *step 5.* on page 47.

8. Continue clicking **Next** and using the hyperlinks to finish annotating the reactions in the pathway. The final screen that displays is the **Save:Finish** screen. You can also click on the **Complete** hyperlink in the left pane to display this screen (*Figure 3.44*) Click **Save** to save the pathway and reactions to either the database or the specified file.

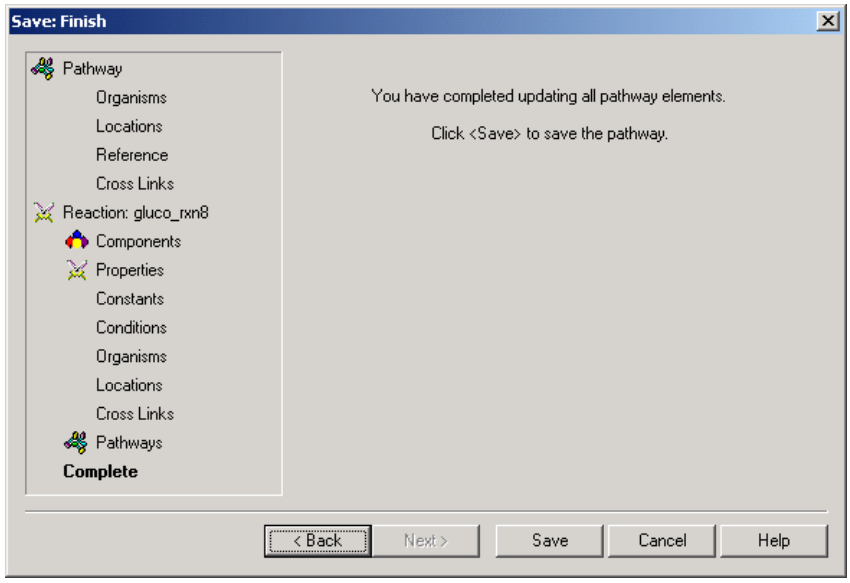


Figure 3.44 Final screen for saving a pathway

Saving Reactions Not Going Through a Pathway

To save reaction(s) independent of a pathway, select the reaction(s), and choose **File > Save Selected Reactions**. The first dialog box that opens is similar to dialog boxes in the preceding section with the exception that only components and properties appropriate to the selected reactions are listed in the left panel (*Figure 3.45*).

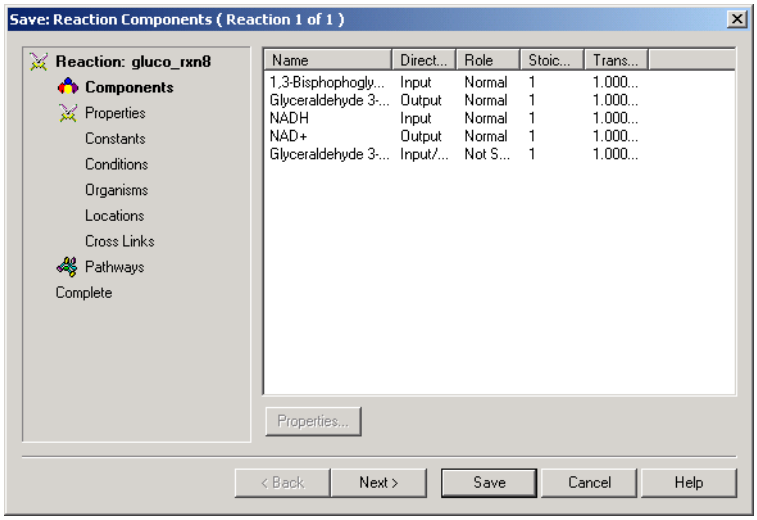


Figure 3.45 In the Save dialog box for saving reactions not going through a pathway, only components and properties of the selected reaction display

The Name column displays each component in the reaction and the remaining columns display information about the connector to which the component is associated in the reaction.

Proceed through the Wizard, as described in the previous section starting with *step 6.* on page 48, entering the information appropriate to the reaction you are saving. Use the **Back** and **Next** buttons for advancing through the screens in sequence.

Note: If you try to change a component that is used in another reaction that is not currently selected, a message displays that “the component is shared with a non-selected reaction and cannot be changed at this time.”

If no reactions are selected, the menu command **Save Selected Reactions** saves all reactions in the Graphics Window. You will still need to step through the Wizard to do so.

Saving a .pw File to the Database

When you open a .pw file and want to save its contents to the database, the following situations can occur:

- None of the objects in the file are already present in the database
- Some of the objects in the file are already present in the database
- All of the objects in the file are already present in the database

Vector PathBlazer determines whether some or all objects in the file are already present in the database and allows you to either use the existing objects in the database or create new objects from the objects in the file. Use the following steps to save the contents of a .pw file to the database.

1. Open the .pw file by selecting **File > Open**, clicking the **Open** button ( ) on the toolbar, or selecting a recently opened .pw file from the list at the bottom of the **File** menu. The Graphics window displays the contents of the file.

2. Select **File > Save** or click the **Save** button ( ) on the toolbar. The **Save** dialog box opens to the screen for naming the pathway. Note that the radiobuttons in the **Save to** box and the **Name** field are not available. Select the **Save as new pathway** checkbox to make these options available.

3. To save the contents of the .pw file to the database, select the **Database** radio button and select a pathway subset from the drop-down list. Name and annotate the pathway and reactions as described in *step 4.* on page 47 through *step 8.* on page 50.

4. When you are finished adding annotations, click **Save**. In the **Save Objects** dialog box the opens (*Figure 3.46*), select:
   - the **Use the existing objects in the database** radio button to use objects that match to the objects in the file by name.
   - the **Save objects with new names** radio button to create new objects in the database from the objects in the file. An incremental number is appended to the newly created version of the object. For example, if ADP is already in the database then a new component called ADP(2) is created.

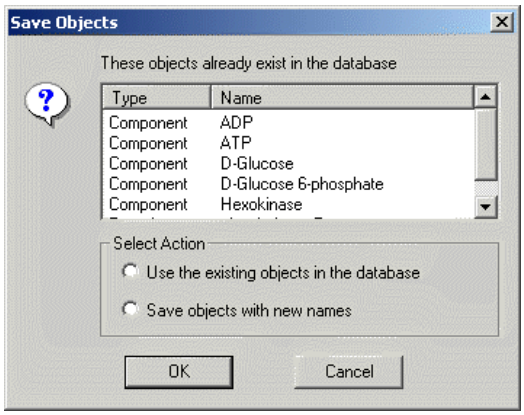


Figure 3.46 Options for saving objects that are already present in the database

Opening Crosslinks to External Databases

In PathBlazer, each component can have two types of crosslinks associated with it: database links and/or URL links.The annotation for describing crosslinks for pathways, reactions, and components allows you to link directly to corresponding objects in the Vector NTI database (if installed) or to defined URLs to obtain additional information such as sequences or citations.

Once crosslinks are assigned as annotations to a particular object, either a crosslink's display name or its literal URL displays in the shortcut menu of a selected object in the Graphics window. The following figure (*Figure 3.47*) shows an object with four crosslinks defined: the first three are URLs to various databases (for example, www.expasy.org/...) and the fourth is the display name to a protein in the Vector NTI database (for example, Interleukin 8 Receptor B). To open any crosslink, click on it in the shortcut menu. If the link is a URL, the default browser opens to the specified page. If the link is to Vector NTI, the viewer opens in the appropriate Vector NTI program.

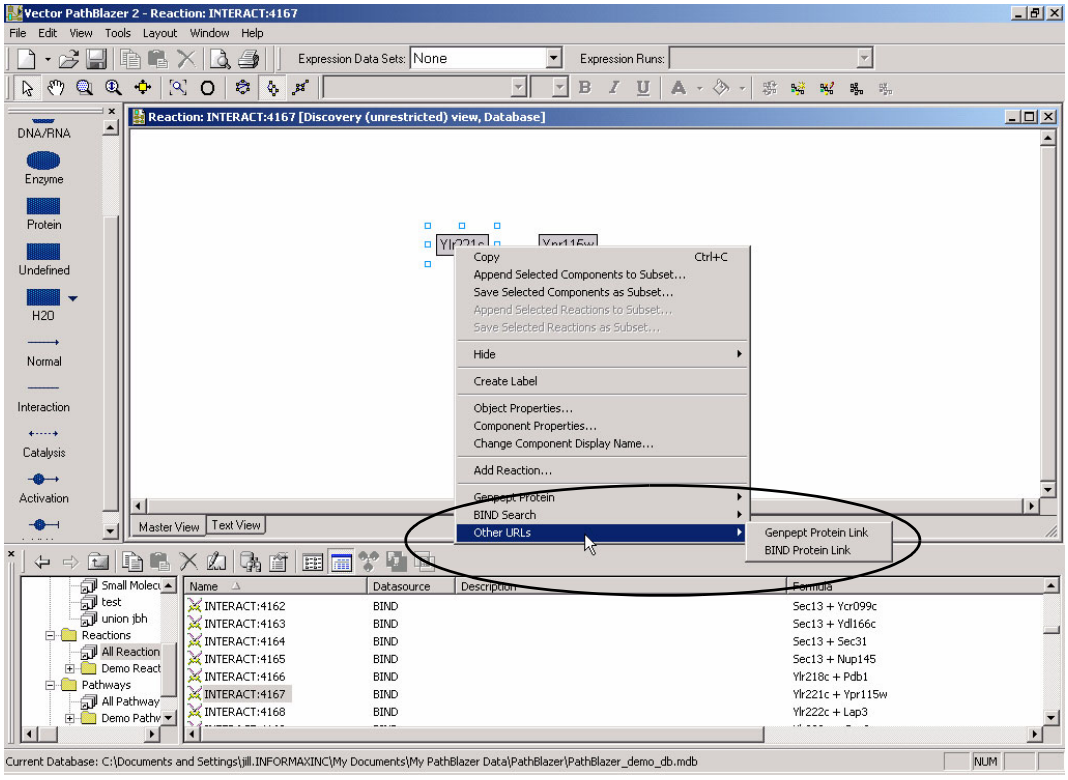


Figure 3.47 Crosslinks display in the shortcut menu of an object in the Graphics window

The G-protein Stimulatory (Gs) pathway that is pre-loaded into the default Vector PathBlazer database is configured with links to corresponding molecules in the VNTI Advance database. The components in the Gs pathway that are linked are:

- Adenylate cyclase to ADCY
- Beta-adrenergic receptor to ADRA1A
- Raf to RAF1
- Phosphodiesterase to PDE1A
- GRK to GPRK2L
- MAPK to MAP2K1
- B-Raf to BRAF

- Epac to EPAC

Searching the Database

There are two ways to search the database for specific objects:

- Search the pathway displayed in the Graphics window for components and reactions by name
- Search the entire database for components, reactions, and pathways by name and/or by annotation. When you search the entire database, you can also create subsets from the search results.

Finding an Object in a Pathway

You might create a pathway that becomes extremely complicated in terms of the numbers of components and reactions, or you might be focusing on a specific part of a pathway and cannot see another part of interest in the same view. To locate a specific component or reaction in a pathway displayed in the Graphics window, use the following steps.

1. Select the window that contains the pathway you want to search. If pathway windows are tiled or cascaded (**Window > Tile** or **> Cascade**), the currently selected window displays a blue title bar. Select **Edit > Find**.
2. In the **Find Pathway Item** dialog box, select either the **Component** or **Reaction** radio button (*Figure 3.48*). When the **Show All** radio button is selected, the list box shows all components or reactions in the pathway by name. To filter components/reactions by name, select the **Show only items containing text** radio button and enter text that matches the items you want to see. Select the component/reaction you want to search for in the List box and click **OK**.

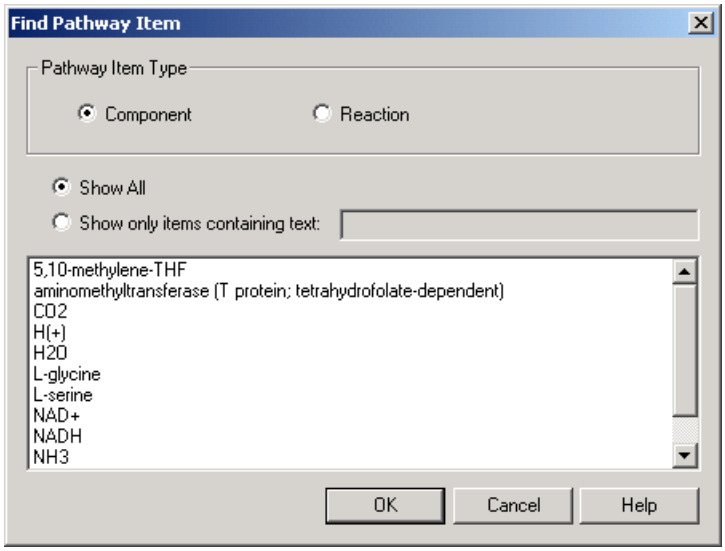


Figure 3.48 Finding a component or reaction in a pathway

3. The component/reaction is centered in the Graphics window and is selected with blue handles (*Figure 3.49*).

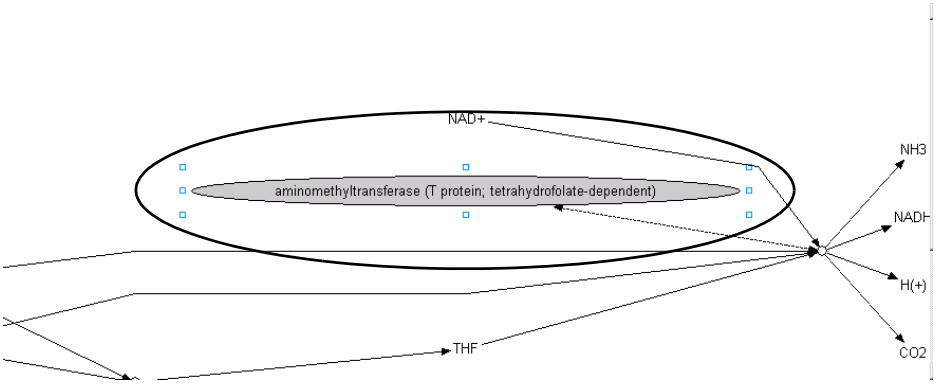


Figure 3.49 Found component is centered and selected in the Graphics window

Searching Objects in the Database and Creating Subsets

An extended search can be performed on all pathways, reactions, and components in the database as well as on annotations that have been added to any objects. Subsets can also be created directly from the search results. To search the database and/or create subsets, use the following steps.

1. Select **Tools > Search Database** or **> Create Subset**. Both commands open the **Search/Create Subset** wizard (*Figure 3.50*). You can also click the **Search** button ( ) in the Explorer toolbar. In the first screen, select the radio button corresponding to the type of object you want to search for and click **Next**.

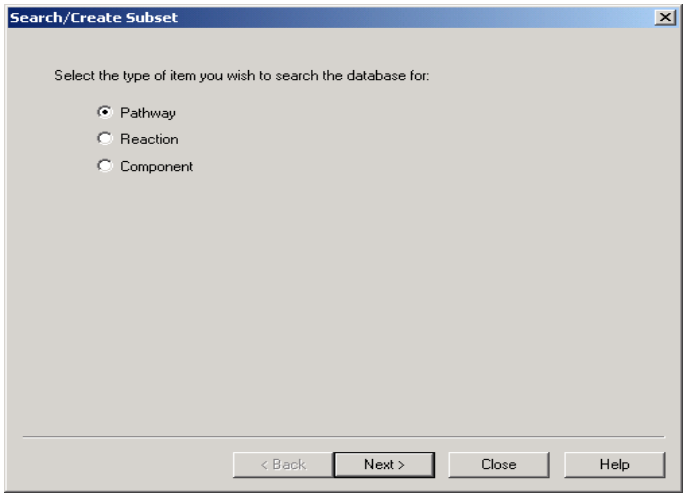


Figure 3.50 Search/Create Subset wizard: selecting a type of object to search for

2. The next screen contains options for configuring one or more search conditions (*Figure 3.51*).
   - Click **Add Single Condition...** to specify a single condition for the search. You can click this button more than one time to add more than one individual condition. To continue with this option, proceed with *step 3*, then move directly to *step 6.* on page 57.
   - Click **Add Multiple Condition...** to specify a set of multiple conditions for the search. You can specify only one set of multiple conditions for a search. One multiple condition can be combined with several single search conditions, however. To continue with this option, proceed with *step 5.* on page 57.

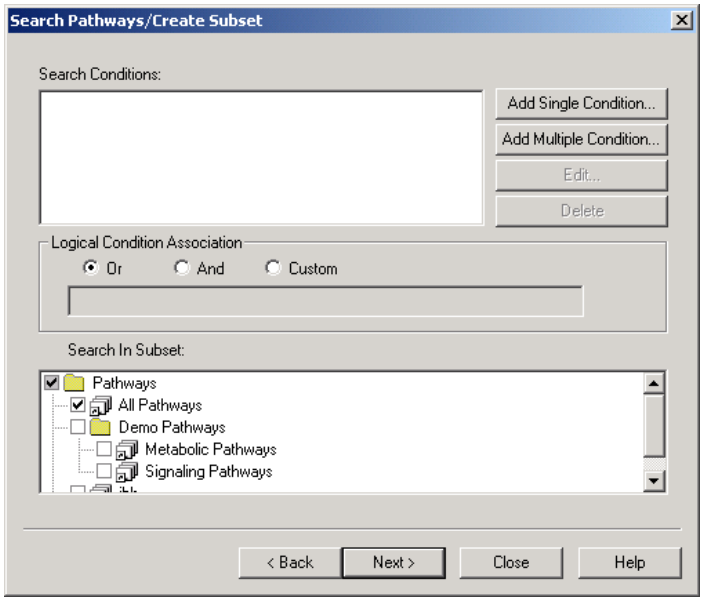


Figure 3.51 Search/Create Subset wizard: configuring a query

3. In the **Add [Single] Condition** dialog box, select a field from the **Condition Type** drop-down list (*Figure 3.52*). The list displays the annotation options for each type of object. Options depend on which kind of object you are searching for. The field that is selected in the **Condition Type** determines the names of additional fields in this dialog box.

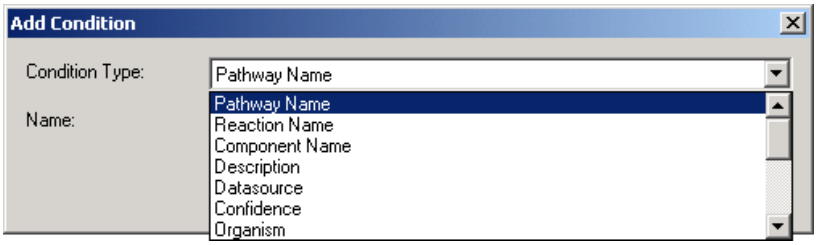


Figure 3.52 Add Condition dialog box

4. Enter an appropriate value in the additional fields or select from a drop-down list of options. For a list of annotations, see *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37.

   The option for **GO annotations** in the **Condition Type** drop-down menu opens a dialog box unique for working with GO annotations. For more information, see *Search Database by GO Annotation* on page 61.

The options for **Location** and **Organism** in the **Condition Type** drop-down menu have two additional conditions for Search Type: **Strict Search** and **Non-Strict Search** (*Figure 3.53*).

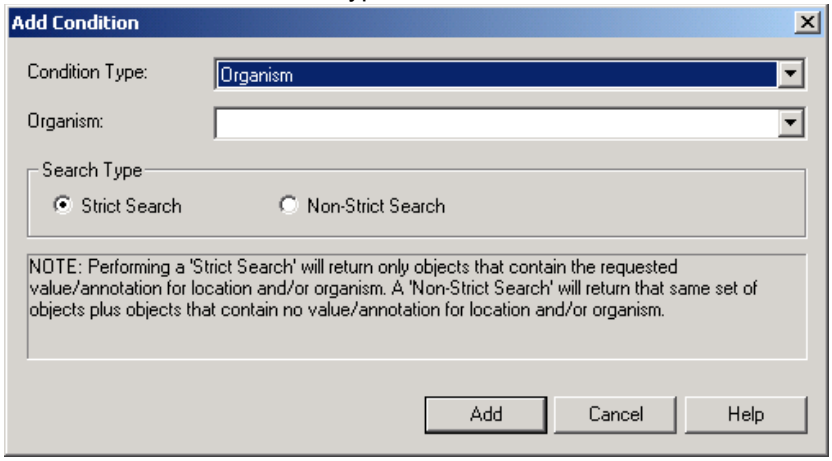

Add Condition
Condition Type:
Organism
Organism:
Search Type
Strict Search
Non-Strict Search
NOTE: Performing a 'Strict Search' will return only objects that contain the requested value/annotation for location and/or organism. A 'Non-Strict Search' will return that same set of objects plus objects that contain no value/annotation for location and/or organism.
Add
Cancel
Help

Figure 3.53 Extra search type options when searching for Location and Organism

These search types, described in the panel below the text boxes, are based on the definitions of **In**, **Known In**, and **Not In**.

- **In** means that an object is definitively known to be in certain organisms or locations only. For example, the protein product for the oncogene ERBA is the ERBA receptor and has been definitively located to the nucleus. Therefore, it is not located anywhere else in the cell. In Vector PathBlazer, the subcellular value for the component ERBA would be **<Location In Nucleus>**.
- **Known In** means that an object is definitively known to be in certain organisms/locations but it cannot or has not been definitively determined whether it is known to be in other organisms/locations. For example, you might definitively determine from a Western blot that ERBA (the ERBA receptor) is present in the nucleus but you cannot experimentally determine whether it is present in the ER. In Vector PathBlazer, the subcellular value for the component ERBA would be **<Location Known In Nucleus>**.
- **Not In** is the opposite of Known In and means that an object is definitively known to not be in certain organisms/locations but it cannot or has not been definitively determined whether it is not known to be in other organisms/locations.

Based on the above definitions of **In**, **Known In**, and **Not In**:

- **Strict Search** means that only objects that are assigned the value of **In** or **Known In** are returned.
  - When an organism/location is assigned the value of **In**, a ***1*** is attributed to that organism/location and a 0 is attributed to all other organism/locations for the purpose of the search.
  - When an organism/location is assigned the value of **Known In**, a ***1*** is attributed to that organism/location and no value is attributed to all other organism/locations.
  - When an organism/location is assigned the value of **Not In**, an ***0*** is attributed to that organism/location and no value is attributed to all other organism/locations.
- **Non-Strict Search** also means that objects that are assigned the value of **In** or **Known In** are returned. **Important:** In a Non-Strict Search, objects that are assigned *no* value for location/organism are also returned.

The following are some example search conditions using **Strict** and **Non-Strict** settings:

- ERBA is in the nucleus. Therefore, it is not in the ER. In Vector PathBlazer, values are set to <nucleus = 1> and <ER = 0>. Both a strict and a non-strict search for <Location =

Nucleus> return ERBA. Both a strict and a non-strict search for <Location = ER> do not return ERBA.

- ERBA is known in the nucleus. Therefore, it is not known if it is in the ER. In Vector PathBlazer, values are set to <nucleus = 1> and <ER = no value>. Both a strict and a non-strict search for <Location = Nucleus> return ERBA. A strict search for <Location = ER> does not return ERBA. However, a non-strict search for <Location = ER> does return ERBA.
- ERBA is in the nucleus and in the ER. In this case, it is definitively known to be in two locations. Even though it is known to be in two locations, it is still not in any other locations. In Vector PathBlazer, values are set to <Nucleus = 1> and <ER = 1>. Both a strict and a non-strict search for either <Location = Nucleus> or <Location = ER> return ERBA. A similar situation occurs if more than one value is known in a location/organism.
- ERBA is not in the nucleus. Again, it is not known if it is in the ER. In Vector PathBlazer, values are set to <Nucleus = 0> and <ER = no value>. Both a strict and a non-strict search for <Location = Nucleus> do not return ERBA. A strict search for <Location = ER> also does not return ERBA. However, a non-strict search for <Location = ER> does return ERBA.

5. In the Add Multiple Condition dialog box, select the **Condition Type** from the drop-down menu (*Figure 3.54*).

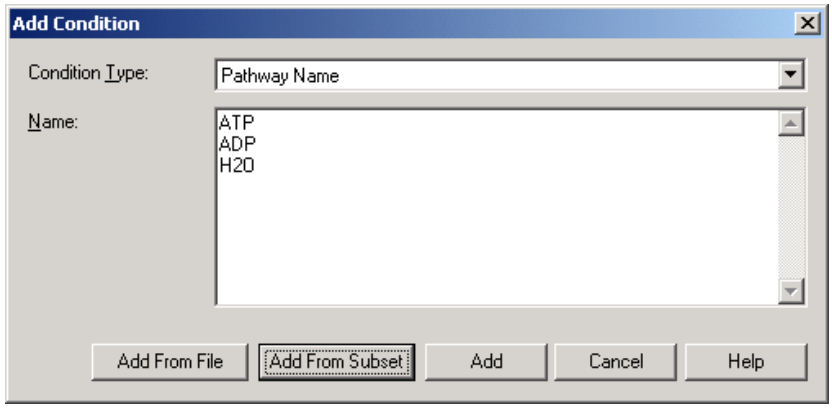


Figure 3.54 The Add Condition dialog box for adding multiple conditions for a database search

In the large text box, add any number of multiple conditions in one of several ways:

- Type the multiple conditions *in list format*.
- Click the **Add from File** button to locate a text file with the search conditions listed.
- Click the **Add from Subset** button to locate an existing subset containing the objects you want to list as conditions. When you choose the subset, then click **Select**; all of the objects in the subset will display in the Add Condition dialog box.

6. When you have finished configuring the search condition(s), click **Add**. The search conditions are added to the Search <object>/Create <object> dialog box with a condition identifier of **C1** next to it. For a single condition, the identifier is specified. For multiple conditions, **C<#> Name = List Condition** displays. To view the specifics of the “List Condition”, select the item, then click the **Edit** button.

Note: You can specify only one set of multiple conditions for a search. (Once you add a multiple condition set, the **Add Multiple Condition** button becomes unavailable.) The conditions making up a multiple condition set are searched with the OR operator. (See the following section, *Custom Search Logic*.) One multiple condition can be combined with several single conditions, however.

- To edit a condition, select it and click **Edit**. (For a multiple condition set, this will display all of the search term values represented by the **Name = List Condition** phrase.)
- To delete a condition, select it and click **Delete**.
- To add additional search conditions, click **Add <single/multiple> Conditions** buttons again, select from the available fields, and enter values.

The condition identifier increases by one with each new condition: C2, C3, etc.

Custom Search Logic

For multiple search criteria, use the **Logical Condition Association** text box to specify the Boolean operator, **AND** or **OR**, that will be used between criteria. See *Figure 3.55*.

- **AND** operator: Only the records that meet both criteria will be returned.
- **OR** operator: Records meeting either search criteria will be returned.

The field below the radio buttons displays the combined query that will be run against the database. For example, C1 and C2 and C3.

Click the **Custom** button for grouping search criteria.

Note: Parentheses are allowed in the Logic text box . Also, you can use a criterion more than once in the Logic field. For example, the expression (#1 AND #2) OR (#1 AND #3) entered in the Login field would find database entries that satisfy *either* criteria #1 and #2 *or* criteria #1 and #3.

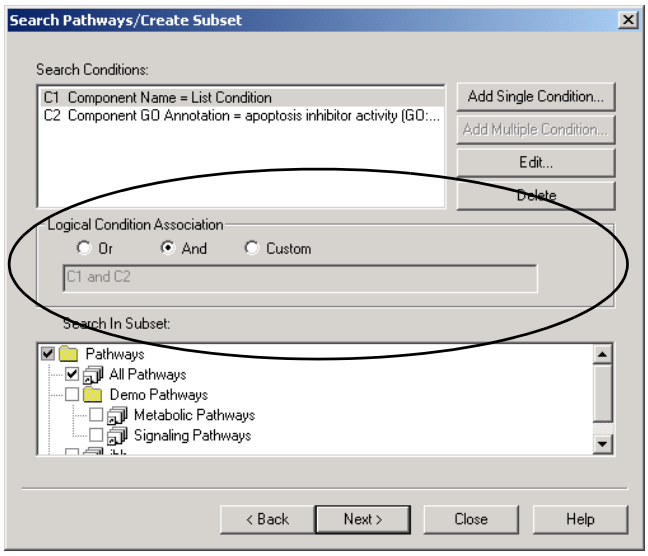


Figure 3.55 Search/Create Subset wizard listing two "logical search conditions"

Note: A text string cannot contain the character '['.

7. Check the checkbox by one or more subsets from the **Search in Subset** folder. Select the **All Reactions/Components/Pathways** subset checkbox to search all database objects of the selected type. Click **Next**.

The search is started. Depending on the search complexity and database size, the search may take several minutes.

Search Results

When the search is complete, the **Search Results** screen lists the objects that meet the search conditions (*Figure 3.57*).

Single Condition Search—The **Name** column lists each returned object by name (*Figure 3.56*). **Description** and **Datasource** columns (for Reactions and Components only) also display values if they have been imported or entered for an object.

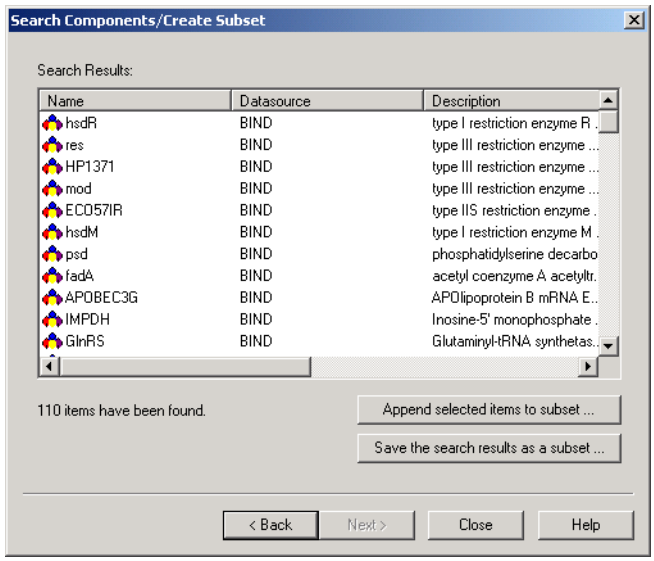

Search Components/Create Subset
Search Results:
Name
Datasource
Description
hsdR BIND type I restriction enzyme R
res BIND type III restriction enzyme
HP1371 BIND type III restriction enzyme
mod BIND type III restriction enzyme
ECO57IR BIND type IIS restriction enzyme
hsdM BIND type I restriction enzyme M
psd BIND phosphatidylserine decarbo
fadA BIND acetyl coenzyme A acetyltr.
APOBEC3G BIND APOlipoprotein B mRNA E.
IMPDH BIND Inosine-5' monophosphate
GlnRS BIND Glutaminyl-tRNA synthetas.
110 items have been found.
Append selected items to subset ...
Save the search results as a subset ...
< Back
Next >
Close
Help

Figure 3.56 Search with Single Condition results

Multiple Condition Search— The **Name** column in the left pane lists the query fields, grouped by search values (*Figure 3.57*). The right-hand panel displays the batch search results, with the number of search terms that were matched for each object found.

You can click on a column header to sort the table by a column's contents. Drag the divider bars to widen or reduce the column widths. View the properties of any object by first selecting it and then selecting **Properties** from the shortcut menu. The Properties dialog box opens, where you can review or change any of the object's properties.

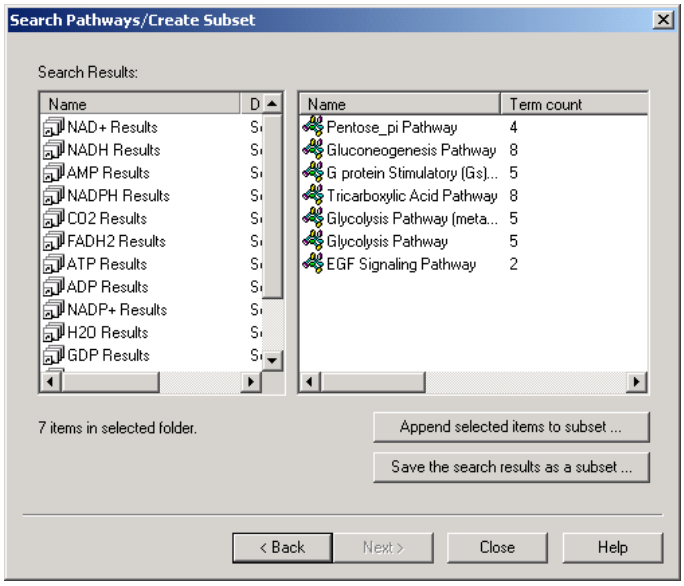


Figure 3.57 Search/Create Subset wizard listing search results

8. There are several options for adding search results to subsets:

Add selected search results to an existing subset—by selecting one or more entries from the list. To select consecutive objects in a list, select the first object, press the SHFT-key, and select the last object. To select non-consecutive objects, press the CTRL-key and select the

objects. Click **Append selected items to subset** to open the **Append to Subset** dialog box (*Figure 3.58*). Click the (**+**) sign to expand the displayed folder (for example, Components), select an existing subset, and click **Append**.

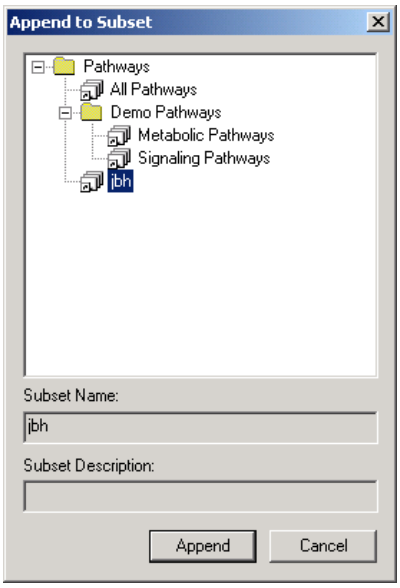


Figure 3.58 Appending search results to an existing subset

You can also copy the selected objects by selecting Copy from the shortcut menu and then paste the copied objects to an existing subset in the Database Explorer by selecting Paste.

Add search results to a new subset—by clicking **Save the search results as a subset**. In the **Create Subset** dialog box, enter a name and description for the subset, and click **Create** (*Figure 3.59*). All the search results listed are saved to the new subset.

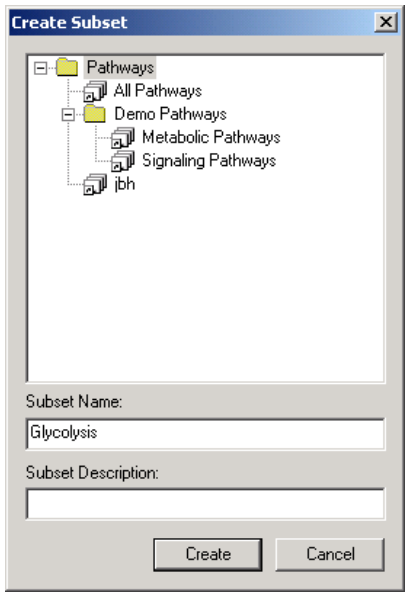


Figure 3.59 Adding search results to a new subset

Note: If the first search does not produce any results, a Select Option dialog box opens at the conclusion of the search, allowing you to select another search option and re-initiate the search.

Search Database by GO Annotation

Note: This search finds objects in the database annotated with GO annotations you specify in the search conditions. The search produces results only after objects in the database have been annotated with GO terms. See *Introduction to Gene Ontologies* on page 153.

To search the database by GO Annotation, you need to initiate the database search as described in *Searching Objects in the Database and Creating Subsets* on page 54. Select the object type in the first dialog box, then select either the **Single** or **Multiple Condition** radio button in the second dialog box. In the Add Conditions dialog box that opens, from the **Condition Type** drop-down list, scroll to the **<object type> GO Annotation** option. From this point on, the search differs from that described in the *Searching Objects in the Database and Creating Subsets* section. Continue as follows:

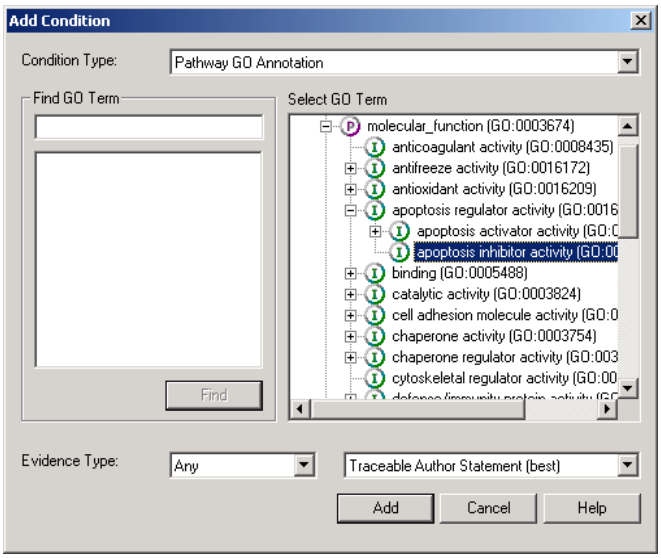


Figure 3.60 The GO Annotation dialog box for adding a GO annotation as a database search condition

In the Add [GO] Condition dialog box that opens, from the GO tree in the right panel, select the GO term you want to set as a search condition (*Figure 3.60*). If you are not sure where in the tree your term is located, enter it in the **Find GO Term** field in the left panel and press the **Find** button. Click on the result in the left panel; the term will be simultaneously highlighted in the GO tree on the right. Click the **Add** button.

The GO term displays as a condition in the Search Pathways/Create Subset dialog box (*Figure 3.61*).

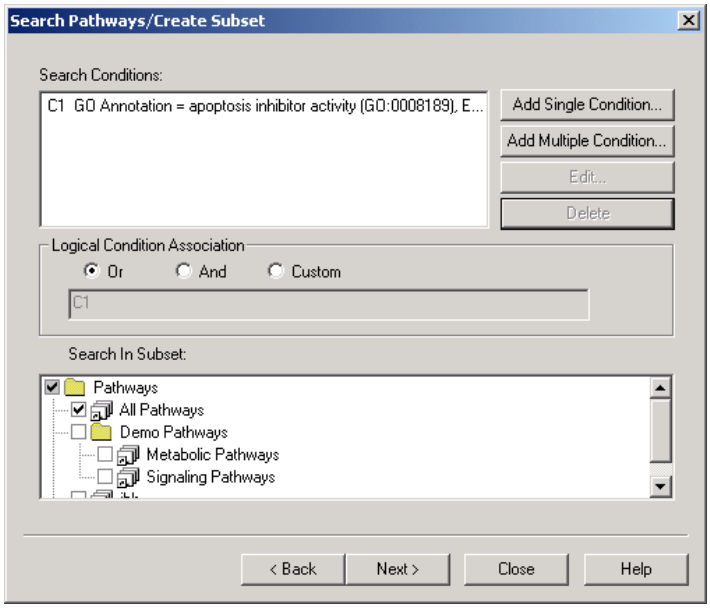


Figure 3.61 The Search <Object>/Create Subset dialog box displaying a GO annotation search condition

In the Search in Subset panel, check one or more subsets to be searched for objects annotated with the GO terms you have set as conditions. Click the **Next** button.

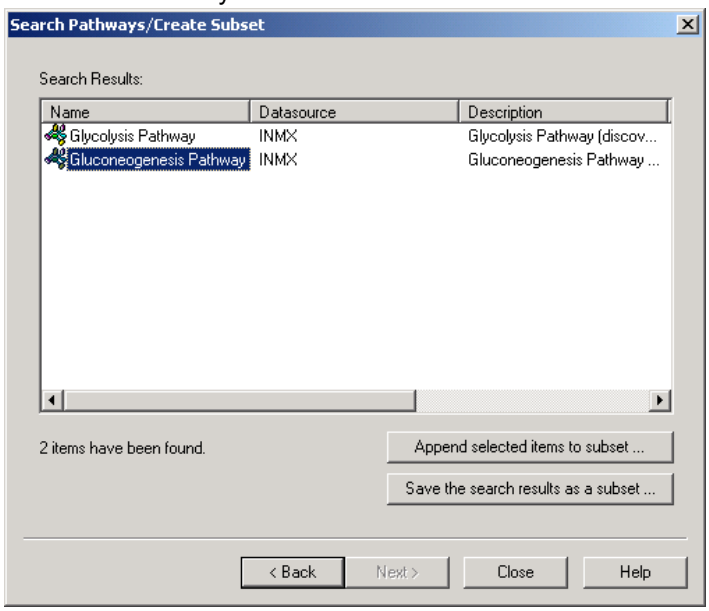


Figure 3.62 Database search results display objects that contain GO annotations used as search conditions

Search results display in the Search Pathways/Create Subset dialog box (*Figure 3.62*). Click the **Back** button to return to modify conditions and re-initiate the search.

View the properties of any object by first selecting it and then selecting **Properties** from the shortcut menu. The Properties dialog box opens, where you can review or change any of the object's properties

To create a subset from the results, select one or more of the result objects and click on of the following buttons:

***Append Selected Items to Subset*—**to save the results as part of an existing database. Select the items you want to save and click the button. In the Append to Subset dialog box, select the subset to store the selected search results and click **Append**.

***Save the Search Results as a Subset*—**to create a new subset containing search results. In the Create Subset dialog box, name and describe the new subset in the appropriate text boxes. Click the **Create** button. This creates a new recipient subset containing all search results and closes the dialog box.

Printing and Saving Images

Publication-ready images can be printed directly from Vector PathBlazer to a local printer. Images can also be saved or copied to the local file system in several common formats, which you can then manipulate using other graphics programs or open in word processing programs.

Printing an Image

Only the contents of the Graphics window can be printed. This includes the contents of the Master View, an Alternate View, or the Text View. The graphical image as it displays in the Graphics window is printed when the Master View or an Alternate View is printed. Any elements that are hidden from view are not printed. In the Text View, any expanded folder and visible element is printed.

For a preview of how a Text View or Master/Alternate View will print, select **File > Print Preview**.

To print the current display, select **File > Print**.

Saving an Image

Only the contents of the Master View or an Alternate View can be saved to an image file. Either all of the contents in the view can be saved to an image file or just the selected contents. Images can be saved in the following file formats:

- JPEG
- Bitmap
- EMF

Use the following steps to save a pathway as an image.

1. To select specific elements in the Graphics window, use the SHFT-key or the CTRL-key to multiple select elements or use the **Select** commands in the **Edit** menu (for example **Edit > Select All Components** selects only the components in the Graphics window).
2. Select **File > Save As Image**.
3. In the **Save As Image** dialog box, select an image format from the drop-down list in the **Type** field (*Figure 3.63*). Click the **Browse** button in the **File Name** field. Navigate to the location where the image will be stored, name the image, and click **OK**.
4. If objects are selected in the Graphics window, then the **Visible Window Only** and the **Selected Objects Only** checkboxes are available in the Image Content field. Otherwise, only the **Visible Window Only** checkbox is available.

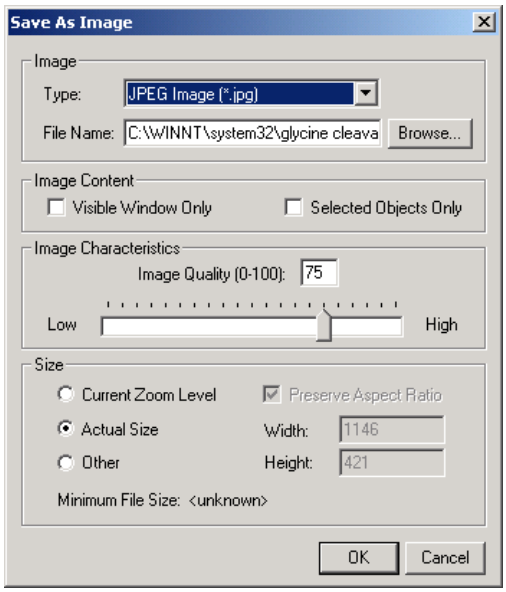


Figure 3.63 Save as Image dialog box

5. In the **Image Characteristics** field, set the image quality by dragging the pointer between Low and High.
6. In the **Size** field, select the size you want the image to be saved in.
7. Click **OK**. The image is saved with the properties you selected in the specified location.

You can also copy any selected elements in the Graphics window to the clipboard and then paste them into a word processing program as a .jpeg image only. To copy an image, use the SHFT-key or the CTRL-key to multiple select elements or use the **Select** commands in the **Edit** menu and then select **Edit > Copy to clipboard** to copy the selected elements to the clipboard.

CHAPTER

4

IMPORTING DATA

This chapter describes how to import public and proprietary data into Vector PathBlazer.

Topics in this chapter include:

- *Introduction to Importing Data* on this page
- *About Vector PathBlazer Data Import* on page 66
- *Importing KEGG Data* on page 72
- *Importing BIND Data* on page 80
- *Importing BioCyc Data* on page 85
- *Importing TransPath Data* on page 93
- *Importing DIP Data* on page 97
- *Importing Proprietary Data* on page 102
- *Pre-Defined URLs* on page 107

For information about importing gene ontologies, see *Introduction to Gene Ontologies* on page 153.

For information about importing expression data, see *Importing Expression Data with a Template* on page 168.

Introduction to Importing Data

One of the strengths of Vector PathBlazer is that it allows importing data from public and proprietary sources, thereby integrating data from different data sources. Public data from the KEGG, BIND, BioCyc, TransPath, and DIP databases can be imported into Vector PathBlazer as well as user PPI and proprietary data.

The general workflow that applies to importing public and proprietary data into Vector PathBlazer is:

1. Public source files are downloaded to the local file system or proprietary files are formatted as XML files according to the Vector PathBlazer Document Type Description (DTD).

2. Source files and other parameters are specified.
3. The program converts public data to XML format and the entries in the source files are imported to create pathway, reaction, and component objects in the database. For proprietary data, which is already in XML format, the program imports the entries in the source files to create objects in the database.

About Vector PathBlazer Data Import

Before you can commence importing data into PathBlazer, you must download the data and store it locally. In some cases, the downloaded files are zipped, and you must unzip them before you can proceed with import. Once you have done so, the PathBlazer Import tool is used for specifying source files, parameters (where appropriate), and importing the data. When you import public data, PathBlazer Import automatically converts the files to Vector PathBlazer XML format for you. When you import proprietary data, you must first format the data in XML format according to the Vector PathBlazer Document Type Definition (DTD) before you can import the data. For information about doing so, see *Appendix B DTD For Data Import.*

Every import session follows the same general steps, no matter what kind of data is being imported.

1. Open the PathBlazer Import [Module] dialog box, where you select the datatype to be imported.
2. Open the Root Folder or Source File dialog box, where you locate and select the root folder or source files of the data.
3. Open the Merge Option dialog box, where you specify how data merge is to be addressed.
4. Execute the data import. A monitor allows you to follow the import process; an import log summarizes the import statistics.

Each part is described in detail in the following subsections, and directions specific for each datatype are described in even more detail in the datatype subsections.

Import Module and Description

Import Module

The **Import Module** field allows you to choose commonly downloaded public or proprietary data sources (*Figure 4.1*). One data type can be imported at a time meaning that if KEGG, for instance, is selected then it is the only data type, public or proprietary, that can be imported in the current import session. Using the scrollbar, choose the supported data types from the available list in the Import Module field.

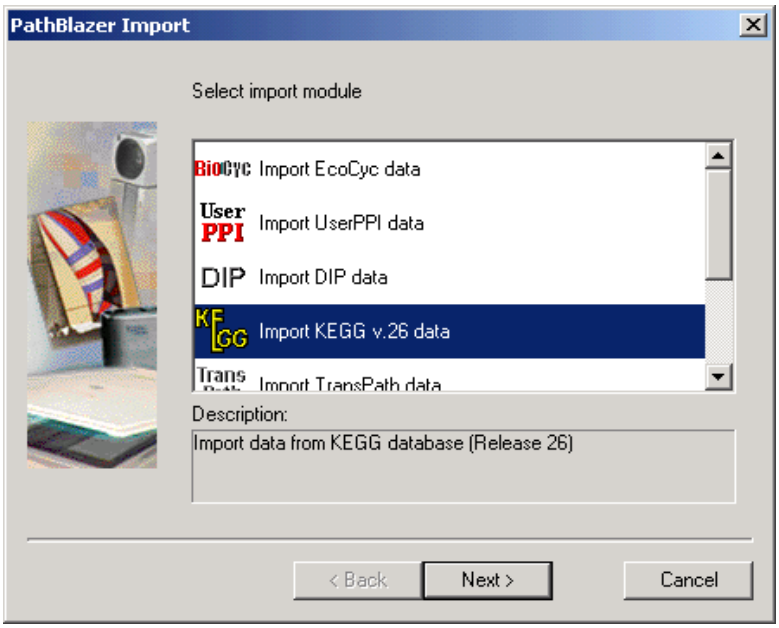


Figure 4.1 PathBlazer Import

Description

The **Description** field identifies the type of data selected for import from the Import Module field. For more information about these databases, see Appendix B, references.

Root Folder or Source File Dialog Box

Each import type utilizes either a root folder or a source file to designate the datasource for import. The dialog box varies according to the datatype being imported. Refer to each datatype subsection for information about using this dialog box.

Root Folder

The **Root Folder** is source folder for data files imported using the KEGG, BioCyc, TransPath and User PPI import tools.

Source File

The **Source File** is the datasource for data imported using DIP, XML, and BIND import tools.

Merge Option Dialog Box

One of the most important features of PathBlazer is data integration. Several different databases can be integrated in one PathBlazer database. This allows you to make cross-database queries, build pathways using data from different sources, find cross talks between metabolic and signal transduction pathways, etc.

To avoid redundancy, data from different databases should be merged, and in PathBlazer, components are merged by default during the database import. Nonetheless, source databases utilize different data models, different substance classifications, etc., and it is inevitable that some components which should be merged will not be, while others will be merged incorrectly.

All merge events are recorded in a log file, described on page 71. If two components are merged incorrectly during import, you can manually re-create a missing component and link it to appropriate reactions. There is no automatic way to 'un-merge' two components.

For more information about merging components manually, see *Merging Components Manually* on page 45.

During import of any new database into the Vector PathBlazer database, the program compares each entry in the source files to entries in the database by its primary name and synonym.

A new component being imported is merged automatically with an old like component only if the name or synonym for the new and old components are identical. If a component has a classification, it can only be merged with a component with the same or deeper level of classification. For example, Component A, classified as 'protein' will be merged with component A, classified as 'protein:regulatory', but not with component A, classified as 'lipid'. In this case, component A 'lipid' will be imported into the PathBlazer database and renamed into ***A (dupl. 1)***.

A Merge Options dialog box opens during every import process, allowing you to define options for merging the data (*Figure 4.2*).

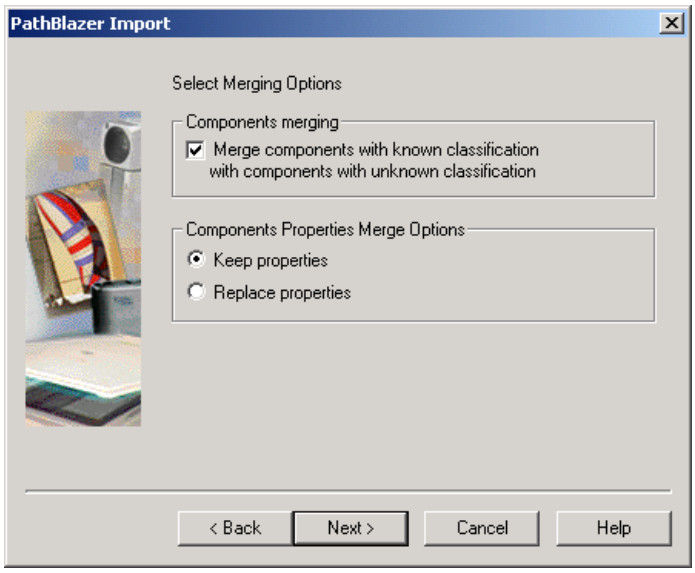


Figure 4.2 PathBlazer Import Merge Option dialog box

You can select the option **Merge components with known classification with components with unknown classification**. If this option is checked, Component A classified as 'protein' will be merged with component A classified as 'unknown'. You can also select a course of action when entries are encountered that are already present in the selected database.

***Keep properties*—**any duplicate entries in the existing database are ignored

***Replace properties*—**any duplicate entries in the existing database are overwritten

If more than one old component matches one new component, they are not merged automatically. Other merge rules for importing components with identical names and different functions then apply.

Merge Component Rules

Components are merged during import if they have the same name AND:

1. For both of them, the Component Class is "unknown".
2. For both of them, the Component Class is the same.
3. If the Component Class of one of them is "unknown" and the option **Merge components with known classification with similarly named components with unknown classification** is selected (*Figure 4.2*).
4. For hierarchical classifications, rules # 2 and #3 are applied recursively.

Two merge examples:

1. A component with some classifications and annotations exists in the PathBlazer database. When you import the next database, it may have some unclassified components, including the one already classified in the PathBlazer database. You can forbid the merge of classified components with unclassified components and hopefully avoid some hard-to-fix mistakes.
2. Components with the same names and classifications DNA > Chromosome > Gene and DNA > Chromosome > Centromere would not be merged. Components with same names and classifications DNA > Chromosome > Gene and DNA > Chromosome > Unknown would be merged only if the option **Merge components with known classification with similarly named components with unknown classification** is selected.

Important: A merge is not executed if a synonym or name, which is used for the merge is less than four symbols. For example, phosphoenolpyruvate that has the synonym PEP (KEGG) will not be merged with Mas 1 from BIND, which also has the synonym PEP. Because, however, source databases utilize different data models, different substance classifications, etc., it is inevitable that some components which should be merged will not be, while others will be merged incorrectly. After import, be sure and review all merge events in the log file, described on page 71.

Note: Some common molecules such as H2O, ATP, and Na+ are always merged.

Merge Results

As a result of a merge, the name and synonyms of a new component are appended to the synonyms of the old component. During the merge process you have the option to retain old attributes or replace them with attributes of the new component.

When the import process is finished, a summary of merged components displays, indicating what was merged and what and how renaming of objects occurred. You can copy this summary to the clipboard and/or save it in a file. This information is also recorded in the log file, described on page 71.

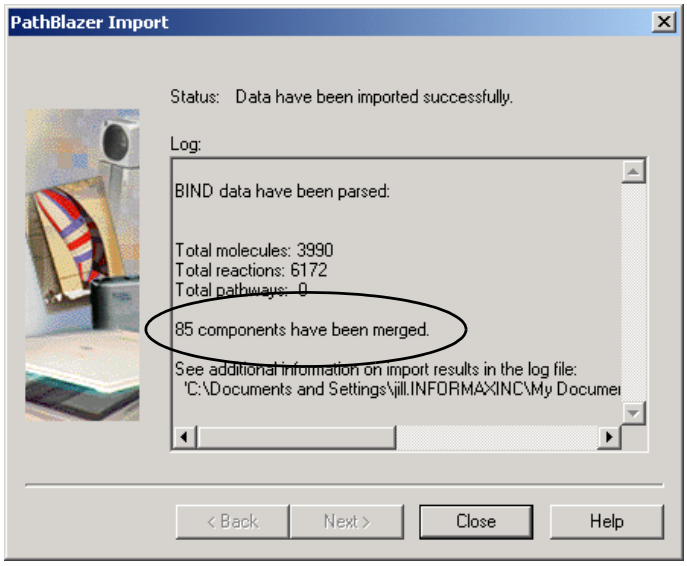


Figure 4.3 PathBlazer import log displaying the number of merged objects

Notes: If components or reactions with the same name do not meet the criteria described and they are not merged, the newly imported component is given a default **dupl.1**, **dupl.2**, etc. meaning duplicate 1, duplicate, 2, etc. or a user specified suffix. If there is already a component or reaction with the same suffix, then the same merge check is applied and an attempt to merge is

made. If it is not successful, an additional numeric suffix is added to the make the component name unique. No further checks are made.

Data can be merged manually after import. For more information, see *Merging Components Manually* on page 45.

Import Session Monitor

The import process proceeds, displaying a monitor to allow you to follow its progress (*Figure 4.4*). When import is finished, an import log displays with a report of the import (*Figure 4.6*).

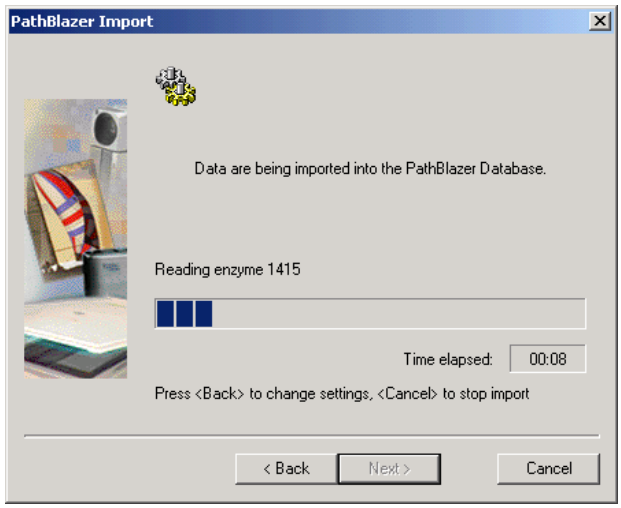


Figure 4.4 PathBlazer import monitor

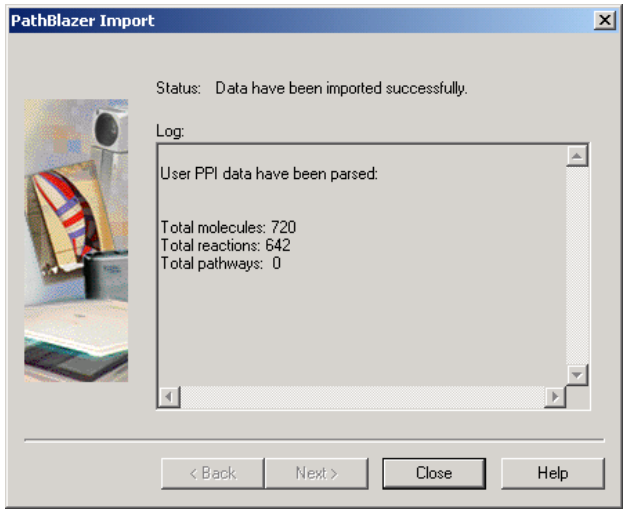


Figure 4.5 PathBlazer import log file

For information about a permanent log file, see *PathBlazer Log File* on page 71.

PathBlazer Import Buttons

The buttons in the PathBlazer Import tool allow you to progress forward, or go back to previous screens to change settings prior to starting an import session, or cancel the import process (Table *4.1*).

Button	*Action*
Back	Reverts the import process back by one screen. To return to the desired screen, continue clicking the **Back** button until the screen of interest is displayed.
Next	Advances the import process to the next screen.
Cancel	Terminates the import process and returns to the user to the Path-Blazer Viewer.

Table 4.1 Buttons and their actions in PathBlazer Import

PathBlazer Log File

A permanent log file, separate from the log file that displays after each import (*Figure 4.3*), is stored in the same folder as the database it was created for. For example: C:\Documents and Settings\<My Documents>\My PathBlazer Data\PathBlazer_demo_db.log.

The log file, a simple text format file, is designed for an advanced user to track and reverse changes of DB objects such as pathways, reactions and components.

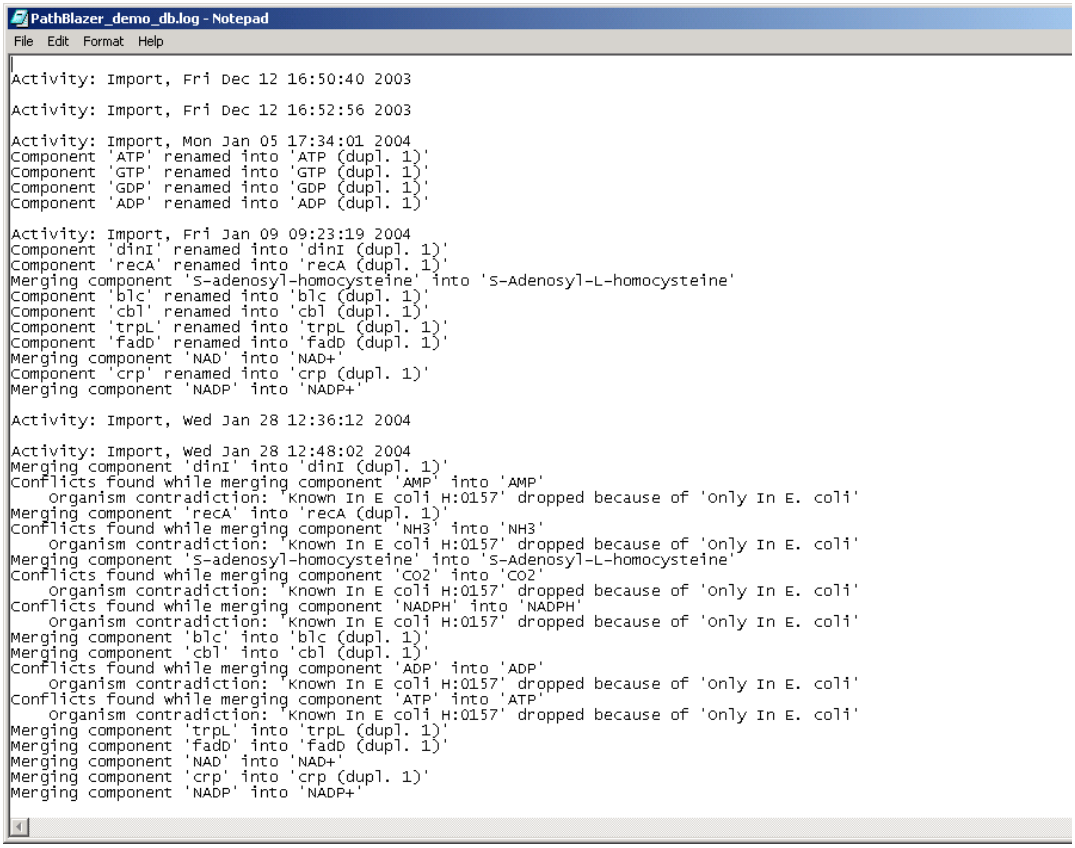

PathBlazer_demo_db.log - Notepad
File Edit Format Help
Activity: Import, Fri Dec 12 16:50:40 2003
Activity: Import, Fri Dec 12 16:52:56 2003
Activity: Import, Mon Jan 05 17:34:01 2004
Component 'ATP' renamed into 'ATP (dupl. 1)'
Component 'GTP' renamed into 'GTP (dupl. 1)'
Component 'GDP' renamed into 'GDP (dupl. 1)'
Component 'ADP' renamed into 'ADP (dupl. 1)'
Activity: Import, Fri Jan 09 09:23:19 2004
Component 'dinI' renamed into 'dinI (dupl. 1)'
Component 'recA' renamed into 'recA (dupl. 1)'
Merging component 'S-adenosyl-homocysteine' into 'S-Adenosyl-L-homocysteine'
Component 'blc' renamed into 'blc (dupl. 1)'
Component 'cbl' renamed into 'cbl (dupl. 1)'
Component 'trpL' renamed into 'trpL (dupl. 1)'
Component 'fadD' renamed into 'fadD (dupl. 1)'
Merging component 'NAD' into 'NAD+'
Component 'crp' renamed into 'crp (dupl. 1)'
Merging component 'NADP' into 'NADP+'
Activity: Import, Wed Jan 28 12:36:12 2004
Activity: Import, Wed Jan 28 12:48:02 2004
Merging component 'dinI' into 'dinI (dupl. 1)'
Conflicts found while merging component 'AMP' into 'AMP'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Merging component 'recA' into 'recA (dupl. 1)'
Conflicts found while merging component 'NH3' into 'NH3'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Merging component 'S-adenosyl-homocysteine' into 'S-Adenosyl-L-homocysteine'
Conflicts found while merging component 'CO2' into 'CO2'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Conflicts found while merging component 'NADPH' into 'NADPH'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Merging component 'blc' into 'blc (dupl. 1)'
Merging component 'cbl' into 'cbl (dupl. 1)'
Conflicts found while merging component 'ADP' into 'ADP'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Conflicts found while merging component 'ATP' into 'ATP'
Organism contradiction: 'Known In E coli H:0157' dropped because of 'Only In E. coli'
Merging component 'trpL' into 'trpL (dupl. 1)'
Merging component 'fadD' into 'fadD (dupl. 1)'
Merging component 'NAD' into 'NAD+'
Merging component 'crp' into 'crp (dupl. 1)'
Merging component 'NADP' into 'NADP+'

Figure 4.6 PathBlazer permanent log file

The file, an example of which is in *Figure 4.6*, is a wrap-around file – when it reaches a certain size limit, the older information gets removed.

The following information is stored in a log file:

- Batch attribute change (when attributes are changed in batch mode, all changes are recorded)
- Import events: component merge and component renaming during import
- Merge of components by manual means after data import

Importing KEGG Data

The KEGG (Kyoto Encyclopedia of Genes and Genomes) database is a collection of interacting molecules and genes based on the current knowledge of molecular and cellular biology. Data in the database is also linked to the gene catalogs produced by genome sequencing projects.[1] A complete description of the contents of the KEGG database as well as licensing information is available at http://www.genome.ad.jp/kegg. Reference and licensing information is also available in *Appendix C*.

KEGG Source Files

KEGG source files are available for download from ftp://ftp.genome.ad.jp/pub/kegg/. This directory has a number of subdirectories including expression, genomes, ligand, pathways, and tar files. Only the Ligand database can be imported into Vector PathBlazer. The Ligand database (Database of Chemical Compounds and Reactions in Biological Pathways) is designed to provide the linkage between chemical and biological aspects of life in the light of enzymatic reactions. The Ligand database is a major component of the DBGET/LinkDB integrated database system (http://www.genome.ad.jp/dbget/), providing useful links among databases such as GenBank and SwissProt.[2]

The Ligand database consists of three parts: the Compound file, the Enzyme file, and the Reaction files. Files are located on the KEGG ftp site in the directory:

ftp://ftp.genome.ad.jp/pub/kegg/ligand/

Download the following files from the ligand directory to a single directory on your local file system. The information below applies to KEGG version 26 and later versions.

- compound
- enzyme
- reaction
- reaction_main.lst
- reaction.lst
- genome

An additional reaction file is required and is located on the same KEGG ftp site in the directory:

ftp://ftp.genome.ad.jp/pub/kegg/ligand/release/20

Download the following file from the release/20 directory to the same directory on your local system:

reaction.main.tar.Z

1. http://www.genome.ad.jp/kegg
2. ftp://ftp.genome.ad.jp/pub/kegg/ligand/ligand.doc

Extract the files from the reaction.main.tar.Z file to a subdirectory in the directory in which the files were downloaded with a file extraction program such as WinZip. Once extracted, numerous .rea files are created in a subdirectory.

A fifth file is needed to assign species names to three-letter species codes in the Enzyme file. Download the following file from the genomes directory to the same directory on your local file system where the other KEGG files were downloaded:

ftp://ftp.genome.ad.jp/pub/kegg/genomes/genome

An example of the contents of each file is given below with an explanation of how the information in the file is parsed into Vector PathBlazer and how it references the data in the other files to create a list of reactions with the corresponding components in the database.

KEGG Import Logic

When the KEGG files listed previously are loaded into Vector PathBlazer, the following steps occur.

1. Each compound listed in the compound file is created in the database as a component.
2. Each enzyme listed in the enzyme file is created in the database as a component.
3. Each compound that is not an enzyme is linked to a reaction from the reaction.lst file.
4. The directionality of reactions is determined from reaction_main.lst file.
5. The formula and name for the reaction are taken from the reaction file.
6. Names of organisms are taken from the genome file.

The result is a set of reactions in the database that each reference the appropriate components. Although KEGG organizes the reactions listed in a .rea file into pathway drawings, Vector PathBlazer does not group these reactions into pathways. Only reactions and components are created from the source files. However, the referenced pathways are preserved in the reactions (Table *4.3*)

Note: Crosslinks from the KEGG have three distinct patterns:

Compounds: db accession C11821

Enzymes: db accession EC 1.7.3.3

Reactions: db accession R00001:EC 3.6.1.10

In summary, a db accession for a compound starts with letter "C", db accession for an enzyme starts with letter "E", and db accession for a reaction starts with letter "R".

KEGG Compound File

The Compound file is a collection of metabolic and other compounds including substrates, products, inhibitors of metabolic pathways, drugs, and xenobiotic chemicals. Each of the chemical substances that appear in the Reaction and Enzyme files and the KEGG/PATHWAY database is identified by an accession number and stored in this file. Each Compound entry contains attribute fields for name, chemical formula, structural formula (in a separate GIF file and a MOL file that cannot be imported into Vector PathBlazer), metabolic pathways, related enzymes, related protein structures, prosthetic groups, and a CAS (Chemical Abstracts Service) registry number[1]. Some of these attributes are imported into Vector PathBlazer as component attributes.

In the Vector PathBlazer database, a separate object (that is, a component of which the type is Undefined) is created for each KEGG compound listed in the file.

1. ftp://ftp.genome.ad.jp/pub/kegg/ligand/ligand.doc

The following is an example of a partial Compound file as it appears in a text editor. Each entry starts with the ENTRY field and ends with the characters '///'. Not all of the fields in the file are imported to the database. The values shown in bold are parsed into an annotation field for the corresponding component. *Table 4.2* contains a mapping of the fields that are extracted by the importer, a field description, and where the value of the field appears in Vector PathBlazer for a component object.

ENTRY C00469
NAME **Ethanol**
Ethyl alcohol
Methylcarbinol
FORMULA **C2H6O**
REACTION R00746 R00754 R02359 R02682 R04410 R05198 R05208
PATHWAY PATH: MAP00010 Glycolysis / Gluconeogenesis
ENZYME 1.1.1.1 1.1.1.2 1.1.1.71 1.1.99.8
...
DBLINKS **CAS: 64-17-5**
///

Field Name	***Description***	***Component Annotation in Vector PathBlazer***
Name	The recommended name of the compound and any alternative names. The recommended name is the first name.	Name: First entry is imported as the primary name to the Name field. This name is the primary name or the unique identifier in Vector PathBlazer. Synonym: All other entries are imported to the Synonym field.
Formula	Chemical formula of the compound	Chemical Formula
DBLinks	Link information to other databases. Currently only contains a link to CAS (Chemical Abstracts Service) and PROMISE (Prosthetic groups and Metal Ions in Protein Active Sites Database) .	CrossLinks Literal value of this field (for example, CAS:"64-17-5) is imported as a crosslink of type 'Database'. A crosslink of type 'URL' is automatically created for all KEGG components. A link is made to the URL http://www.genome.ad.jp/dbget-bin/www_bget?compound+<entry> where <entry> is the value of the ENTRY field in the component file A list of pre-defined URLs is listed in *Pre-Defined URLs* on page 107.

Table 4.2 Imported attribute fields in the Compound file

The ID number in the **Entry** field is used to link a reaction to a component but is not actually parsed into Vector PathBlazer. This is described in further detail in *KEGG Reaction Files* on

page 77. Also, the attribute **Datasource** is automatically defined as **KEGG** for all imported KEGG components.

KEGG Enzyme File

The Enzyme file is a collection of all known enzymatic reactions classified according to the nomenclature of the International Union of Biochemistry and Molecular Biology (IUBMB). Each Enzyme entry is identified by an EC (Enzyme Commission) number and contains attribute fields for name, reaction, metabolic compounds, metabolic pathways, genes encoding the enzyme for several organisms (mainly completely sequenced ones), genetic diseases, and links to other databases including protein sequence motifs and 3D structural data.[1] Some of these attributes are imported into Vector PathBlazer as component attributes.

In the Vector PathBlazer database, a separate object (that is, a component of which the type is Enzyme) is created for each KEGG enzyme listed in this file.

The following is a partial example of the Enzyme file as it appears in a text editor. The values in bold are the values that are imported into Vector PathBlazer. Table *4.3* contains a mapping of the fields that are extracted by the importer, a field description, and where the value of the field appears in Vector PathBlazer for a component object.

ENTRY	**EC 1.1.1.1**
NAME	**alcohol dehydrogenase** **alcohol reductase**
CLASS	Oxidoreductases
SYSNAME	**alcohol:NAD oxidoreductase**
REACTION	an alcohol + NAD = an aldehyde or ketone + NADH2 ...
SUBSTRATE	alcohol NAD
PRODUCT	NADH ketone aldehyde
COFACTOR	Zinc
COMMENT	**A zinc protein. Acts on primary or secondary alcohols or hemi-acetals; the animal, but not the yeast, enzyme acts also on cyclic secondary alcohols.**
REFERENCE	1 Branden, G.-I., Jornvall, H., Eklund, H. and Furugren, B. Alcohol dehydrogenase. In: Boyer, P.D. (Ed.), The Enzymes, 3rd ed., vol. 11, Academic Press, New York, 1975, p. 103-190. ...
PATHWAY	**PATH: MAP00010 Glycolysis / Gluconeogenesis** **PATH: MAP00071 Fatty acid metabolism**
GENES	**HSA: 124(ADH1A) 125(ADH1B) 126(ADH1C) 127(ADH4) 128(ADH5) 130(ADH6) 131(ADH7)** ...
DISEASE	**MIM: 103700 Alcohol dehydrogenase IA (class I), alpha polypeptide** ...
MOTIF	**PS: PS00059 G-H-E-x(2)-G-x(5)-[GA]-x(2)-[IVSAC]** ...

1. ftp://ftp.genome.ad.jp/pub/kegg/ligand/ligand.doc

STRUCTURES **PDB: 1A4U 1A71 1A72 1ADB 1ADC 1ADF 1ADG 1AGN 1AXE 1AXG**
...

DBLINKS **IUBMB Enzyme Nomenclature: 1.1.1.1**
ExPASy - ENZYME nomenclature database: 1.1.1.1

Field Name	***Description***	***Component Annotation in Vector PathBlazer***
Entry	EC (Enzyme Commission) number	Component Class/EC Number Also appended to the reaction name Example: R00754:EC 1.1.1.1
Name	Recommended name and any alternative names of the enzyme	Component Name First entry is imported as the primary name and is appended with the E.C. number. This becomes the unique identifier in Vector PathBlazer. Component Synonym All other entries are imported as synonyms
SysName	Systematic name given by the Enzyme Commission, which represents the nature of the chemical reaction	Component Synonym
Comment	Text information about the enzyme	Component Description
Genes	Link information to KEGG gene catalogs 3-letter organism abbreviation is followed by the list of genes that encode the enzyme. For a key to the abbreviations, see ftp://ftp.genome.ad.jp/pub/kegg/ligand/ligand.doc. The 3-letter abbreviations are defined by links made to the Genome file.	Component Organism
Disease	Link information to disease descriptions in OMIM (Online Mendelian Inheritance in Man)	Component CrossLinks A pre-defined URL that matches the OMIM database is automatically defined. Pre-defined URLs are listed in *Pre-Defined URLs* on page 107.
Motifs	Link information to motif definitions in the Prosite database	Component CrossLinks A pre-defined URL that matches the Prosite database is automatically defined. Pre-defined URLs are listed in *Pre-Defined URLs* on page 107

Table 4.3 Imported attribute fields from the Enzyme file

Field Name	*Description*	*Component Annotation in Vector PathBlazer*
Structures	Link information to 3-D protein structures in PDB (Protein Data Bank)	Component CrossLinks A pre-defined URL that matches the PDB database is automatically defined. Pre-defined URLs are listed in *Pre-Defined URLs* on page 107.
DBLinks	Link information to other databases including: IUBMB Enzyme Nomenclature ENZYME Nomenclature database at Swiss Institute of Bioinformatics WIT (What Is There) Interactive Metabolic Reconstruction on the Web UM-BBD (Biocatalysis/Biodegradation Database) BRENDA SCOP (Structural Classification of Proteins).	Component CrossLinks A crosslink of type 'URL' is automatically created for all KEGG enzymes. A link is made to the URL http://www.genome.ad.jp/dbget-bin/www_bget?enzyme+<EC Number> where <EC Number> is the value of the ENTRY field in the Enzyme file Pre-defined URLs are listed in *Pre-Defined URLs* on page 107.
Pathway	Reference to KEGG maps	Reaction Pathway

Table 4.3 Imported attribute fields from the Enzyme file (Continued)

KEGG Reaction Files

The Reaction files are a collection of chemical reactions that appear in the pathway diagrams of the KEGG/PATHWAY database as well as in the Enzyme file. Reactions include non-enzymatic reactions and enzymatic reactions whose E.C. numbers have not been assigned yet.

There are three kinds of Reaction files: reaction, reaction.lst, and reaction.main.lst. All files include chemical equations.

Reaction.lst file

The file reaction.lst lists all reactions appearing in the Enzyme file and the KEGG/Pathway database. Each line corresponds to a separate reaction and is given a unique ID. Each reaction entry starts with the reaction ID followed by a colon ':' followed by the reaction written as a chemical equation. The identification numbers in the chemical equation reference values in the Entry field in the Compound file. The following is a partial example of the reaction.lst file as it appears in a text editor.

```
R00702: 2 C00448 <=> C00013 + C03428 + C00080
...
```

A reaction object is created in the Vector PathBlazer database for each KEGG reaction using the reaction ID in the reaction.lst file. Reactions are then linked to components by matching a component ID from the chemical formulas field in the reaction.lst file to the corresponding val-

ues in the entry field of the Compound file. For example, the chemical formula in reaction R00702 is:

2 C00448 <=> C00013 + C03428 + C00080

C00448 matches the corresponding record in the Compound file:

ENTRY C00448

NAME trans,trans-Farnesyl diphosphate
Farnesyl diphosphate
Farnesyl pyrophosphate
2-trans,6-trans-Farnesyl diphosphate

...

Thus, reaction **R00702** is linked to component **C00448**, which is **trans,trans-Farnesyl diphosphate** by primary name in the Vector PathBlazer database. Other components in this reaction will be linked: **C00013** to **Pyrophosphate**; **C03428** to **Presqualene diphosphate**, and **C00080** to **H+**. Component **C00448** in this will have stoichiometric coefficient **2**.

Reaction_main.lst File

This file is used to determine directionality of reactions. For example,

R00093: C00025 <= C00064 + C00026

R00094: C00051 <=> C00127

...

R00093 will be directed from right to left. If, during import, the checkbox **Create and store reverse reactions for reactions of known directionality** is checked the reverse reaction

R00093-{Reverse}: C00064+C00026=>C00025 will also be created.

For **R00094** direct and reverse reactions will be created by default.

Reaction file

This file is used to assign a Name and Formula to the reaction. For example

ENTRY R00093

NAME L-Glutamate:NAD+ oxidoreductase (transaminating)

DEFINITION 2 L-Glutamate + NAD+ <=> L-Glutamine + 2-Oxoglutarate + NADH

...

will be named "L-Glutamate:NAD+ oxidoreductase (transaminating)" and have the formula

2 L-Glutamate + NAD+ <=> L-Glutamine + 2-Oxoglutarate + NADH.

KEGG Genome File

The Genome file contains information about completely sequenced organisms. This file is used by the Vector PathBlazer importer to assign a species to the three-letter species codes listed in the Gene field in the Enzyme file. For example, in the partial example of the Enzyme file shown

in the *KEGG Enzyme File* on page 75, the GENES field contains the entry HSA:124(ADH1A) .... HSA in the GENES field of the Enzyme file is matched to the corresponding value in the ENTRY field in the Genome file. The species is then determined by assigning the value in the DEFINITION field in the Genome file to the Organism attribute in Vector PathBlazer. In this example, the species is identified as Homo sapiens. A partial example of the Genome file as it appears in a text editor follows. The fields that are referenced are in bold.

ENTRY	**hsa**
NAME	**H.sapiens**
DEFINITION	Homo sapiens
TAXONOMY	TAX:9606
LINEAGE	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo

...

///

Instructions for Importing KEGG

You can import data either into the default PathBlazer database or into a new separate database you create before data import. To create a database, see *Creating a New Database* on page 10. You must also have downloaded the data files described in *KEGG Source Files* on page 72 to your local file system.

Use the following steps to import KEGG data into Vector PathBlazer.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. Open PathBlazer Import by selecting **File > Import**. PathBlazer Import opens displaying the various import options (*Figure 4.7*).

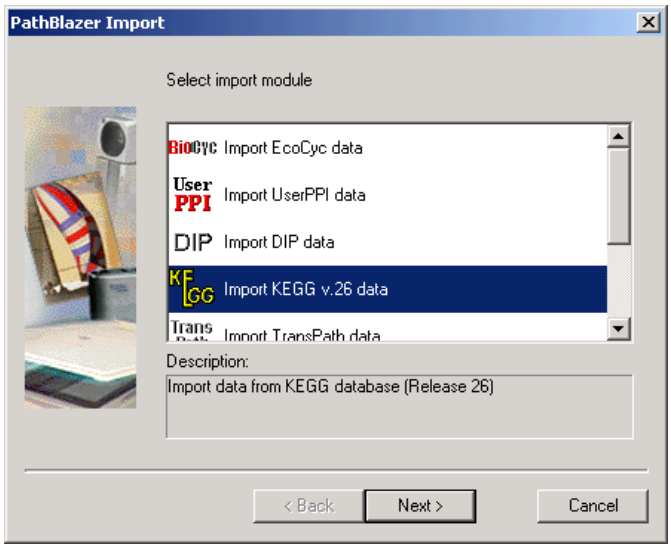


Figure 4.7 PathBlazer Import displaying KEGG v.26 settings

3. In the Select Import Module box, choose **Import KEGG v.26 Data**. The **Description** box reflects the type of data chosen for import. Click **Next**.
4. In the **KEGG Settings** dialog box, select the KEGG data directory where you downloaded the source files you previously created by clicking the **Browse** button adjacent the **Root**

Folder field. Locate the corresponding root folder in the **Browse for Folder** dialog box, and click **OK**. The complete path to the folder displays in Root Folder field (*Figure 4.8*).

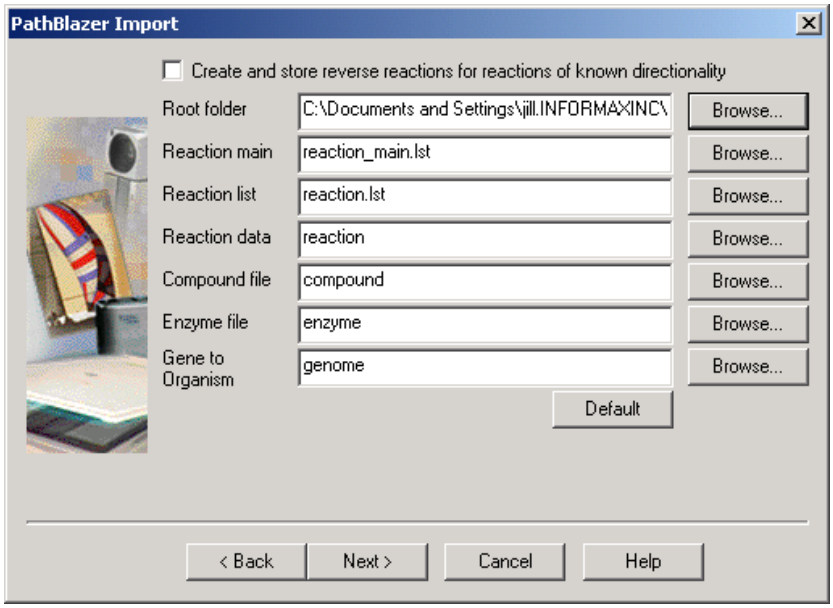


Figure 4.8 KEGG files selected for import

Continue to select the data file for import by clicking the **Browse** button adjacent to each field, locating the corresponding file in the **Browse for File** dialog box, and clicking **Open**. The complete path to the file displays in each field.

- *Optional*: Select the **Create and store reverse reactions...** checkbox. If this box is selected, it creates reverse reactions for reactions with a KNOWN directionality.
- *Optional*: Select the **Save intermediate XML file** checkbox to save the XML file that is created from the KEGG source files. Specify a location and a file name for the XML file by clicking the **Browse** button adjacent to the field **Select path for intermediate XML** file.

5. When you are finished selecting the KEGG directory and files, click **Next**.
6. In the **Merge Option** dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information.
7. To import the data, click **Next**. The data loads while a monitor displays, allowing you to follow the import process. An import log summarizing import results displays when the import has been successfully complete. To stop the import, click **Cancel**. A message displays when the import is complete. Click **Close**.
8. Once imported, verify the import process by choosing an example of a KEGG reaction in the Graphics window.

Note: Reaction directionality is explicit in the KEGG database (v.22 and later). If the reaction is bi-directional, it is stored as two reactions in PathBlazer.

Importing BIND Data

BIND (Biomolecular Interaction Network Database) stores full descriptions of interactions, molecular complexes, and pathways. Development of the BIND 2.0 data model has led to the incorporation of virtually all components of molecular mechanisms including interactions between any two molecules composed of proteins, nucleic acids and small molecules. Chemical reactions, photochemical activation, and conformational changes can also be described. The

database can be used to study networks of interactions, to map pathways across taxonomic branches and to generate information for kinetic simulations.[1]

A complete description of the contents of the BIND database as well as licensing information is available at http://www.binddb.org/. Reference and licensing information is also available in *Appendix C*.

BIND Source Files

Three main data types are defined in the BIND database:

- **Interactions**: contain two BIND objects. A BIND object describes a molecule of any type.
- **Molecular complexes**: define and describe the interactions between any two molecules. The majority of stored information is between proteins, DNA, and RNA.
- **Pathways**: define collections of more than two interactions.

Each object is composed of various component and descriptive objects, which can be imported into Vector PathBlazer as annotations.

The data that can be imported into the Vector PathBlazer database is included in three division files. One or more of the files are available for download, one by one, at ftp://ftp.bind.ca/pub/BIND/DB/archive/. These files contain information about components and reactions.

Download the **BIND_Interaction.xml.gz file** to your local file system and then extract the file with a program such as WinZip. Once extracted, the file **BIND_Interaction.xml** is created.

A partial example of the contents of the BIND_Interaction.xml file is shown below with an explanation of how the information in the file is parsed into Vector PathBlazer. A list of reactions with the corresponding components is created in the database from the file. The BIND Document Type Definition (DTD) can be found at ftp://ftp.bind.ca/BIND/Spec/xmldtd/. The values that are directly parsed are shown in bold.

XML Source:

```
...
<BIND-Interaction>
…
    <BIND-Interaction_iid>
      <Interaction-id>301</Interaction-id>
    </BIND-Interaction_iid>
    <BIND-Interaction_a>
      <BIND-object>
        <BIND-object_short-label>Ade2
        </BIND-object_short-label>
        <BIND-object_other-names>
          <BIND-object_other-names_E>O3293
          </BIND-object_other-names_E>
          <BIND-object_other-names_E>YOR3293
          </BIND-object_other-names_E>
          <BIND-object_other-names_E>YOR128C
          </BIND-object_other-names_E>
        </BIND-object_other-names>
  …
        <BIND-object_origin>
```

1. http://www.binddb.org/

```
          <BIND-object-origin>
            <BIND-object-origin_org>
              <BioSource>
                <BioSource_org>
                  <Org-ref>
                    <Org-ref_taxname>Saccharomyces cerevisiae</Org-ref_taxname>
        …
                  </Org-ref>
                </BioSource_org>
              </BioSource>
            </BIND-object-origin_org>
          </BIND-object-origin>
        </BIND-object_origin>
OR
<BIND-object_origin>
   <BIND-object-origin>
     <BIND-object-origin_chem>
       <BIND-chemsource>
         <BIND-chemsource_names>
   <BIND-chemsource_names_E>
    LY294002
   </BIND-chemsource_names_E>
   <BIND-chemsource_names_E>
    2-(4-Morpholinyl)-8-phenyl-4H-1-benzopyran-4-one
   </BIND-chemsource_names_E>
   <BIND-chemsource_names_E>
    2-(4-morpholinyl)-8-phenochrome
   </BIND-chemsource_names_E>
           </BIND-chemsource_names>
           <BIND-chemsource_chemical-formula>
            C19H17NO3
           </BIND-chemsource_chemical-formula>
   …
       </BIND-chemsource>
     </BIND-object-origin_chem>
    </BIND-object-origin>
   </BIND-object_origin>
   …
     </BIND-object>
    </BIND-Interaction_a>
    <BIND-Interaction_b>
     <BIND-object>
      <BIND-object_short-label>Ade2</BIND-object_short-label>
   …
     </BIND-object>
    </BIND-Interaction_b>
   …
   </BIND-Interaction>
```

BIND Import Logic

Each BIND interaction is defined by the tag <BIND-Interaction>. Each interaction is made up of two components stored between the tags <BIND-Interaction_a> and <BIND-Interaction_b>. A reaction object is created in the Vector PathBlazer database for each interaction listed in the file. Component objects are created from each component stored in an interaction. The following table describes each XML attribute or element for which a value is directly parsed and the annotation to which it is mapped to an object in the program.

XML tag	***Description***	***Annotation in Vector PathBlazer***
<Interaction-id> </Interaction-id>	Interaction name	Reaction Name. ID is appended with the text 'Interact:'. For example, Interact:301 Reaction Crosslink. ID is appended with the text 'BIND:INTERACT' For example, BIND:INTERACT:301 **Note:** Components and reactions including components named "UNDEFINED", "UNKNOWN", "-", "Homo sapiens" or an empty value are skipped.
<BIND-object_short-label> </BIND-object_short-label>	Short label of the object Example: ATP, S4, HSP70	Component Name
<BIND-object_other-names_E> </BIND-object_other-names_E>	Synonyms	Component Synonyms
<Org-ref_taxname> </Org-ref_taxname>	Species	Component Organism
<BIND-chemsource_chemical-formula>	Chemical formula	Component Chemical Formula
<BIND-chemsource_names_E>	Chemical names	Component Synonyms
<BIND-object-type-id_protein> or <BIND-object-type-id_dna> or <BIND-object-type-id_rna> or <BIND-object-type-id_small-molecule> or <BIND-object-type-id_complex> or <BIND-object-type-id_gene> or <BIND-object-type-id_photon>	Component type	Component Type or Subtype
<Geninfo-id>	Link to GI database	Component CrossLink
<BIND-other-db_dbname> and <BIND-other-db_strp>	Link to other databases for small molecules	Component CrossLink
<BIND-cellstage_phase>	Cell cycle phase	Component Locations
<BIND-gen-place>	General cellular location where an interaction takes place	Component Locations

Table 4.4 XML attributes and elements in the BIND_interaction.xml file that are imported

XML tag	Description	Annotation in Vector PathBlazer
<BIND-membrane>	Description of a location in a lipid bilayer membrane	Component Locations
< BIND-path-descr_descr>	Description of the compo-nents	Component Comments
<BIND-descr_simple-descr>	Description of the reaction	Reaction Comments

Table 4.4 XML attributes and elements in the BIND_interaction.xml file that are imported (Continued)

A list of pre-defined URLs that are automatically setup for BIND components and reactions is listed in *Pre-Defined URLs* on page 107.

Instructions for Importing BIND

BIND currently has several "division" databases, each comprising a separate .xml file. Each file must be imported separately.

You can import data either into the default PathBlazer database or into a new separate database you create before data import. To create a database, see *Creating a New Database* on page 10. You must also have downloaded the data file described in *BIND Source Files* on page 81 to your local file system.

Use the following steps to import BIND data into Vector PathBlazer.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. With PathBlazer open, select **File > Import**. The PathBlazer Import tool opens, displaying the various import options (*Figure 4.9*).

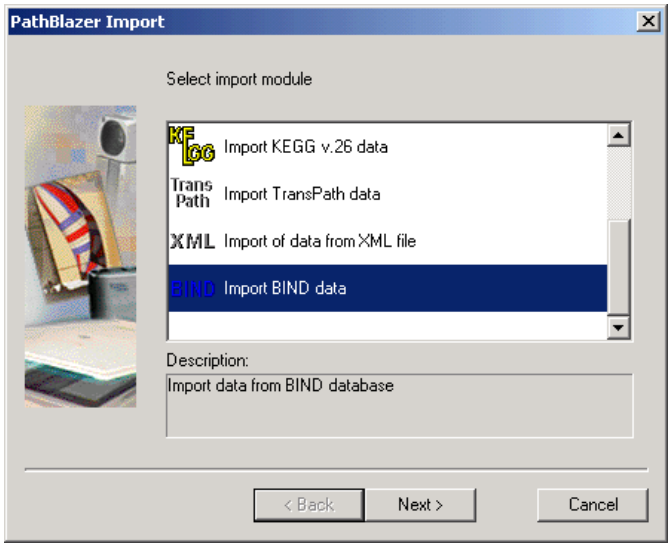


Figure 4.9 PathBlazer Import selecting the BIND import option

3. In the Select Import Module screen of the Import Wizard, choose **Import BIND Data**. The Description box reflects the type of data chosen for import. Click **Next**.

4. In the **BIND Settings** dialog box, select the BIND_Interaction.xml file for import by clicking the **Browse** button. Locate the corresponding file in the **Open** dialog box, and click **Open**. The complete path to the file displays in the **Select source file** field (*Figure 4.10*).

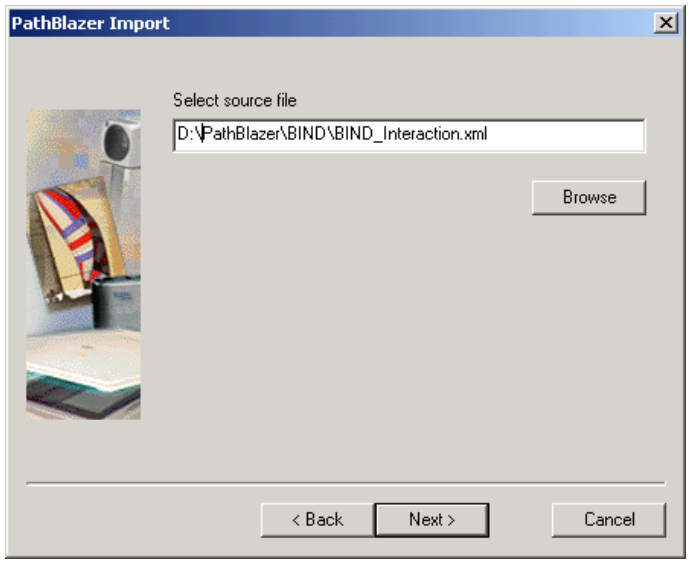


Figure 4.10 BIND file selected for import

5. Click **Next**.
6. In the **Merge Option** box (see *Figure 4.2*), select the merge options. For more information, see *Merge Option Dialog Box* on page 67.
7. To import the data, click **Next**. The **Load BIND** dialog box opens and displays the progress of the import. To stop the import, click **Cancel**.
8. A message displays when import is successfully completed. Click **Close**.

Since BIND data only contains information about interactions between two components, there are no predicted products and each BIND interaction is represented as a one-sided equation.

Importing BioCyc Data

BioCyc, at the time of this writing, is a collection of 17 bioinformatics databases that describe the genome and the characterized biochemical machinery of model organisms whose entire genomes have been sequenced, such as *Escherichia coli*, *Homo sapiens*, and *Agrobacterium tumifaciens.* For instance, in the case of *Escherichia coli*, the EcoCyc database describes the mechanisms of transcriptional regulation of *E. coli* genes, and contains the complete genome sequence of *E. coli*, and describes the nucleotide position and function of every *E. coli* gene. EcoCyc also describes *E. coli* operons, promoters, transcription factors, and transcription-factor binding sites.

A complete description of the contents of BioCyc, licensing information as well as downloadable databases are available at www.biocyc.org and http://biocyc.org/flat-file-reg.shtml . Reference and licensing information is also available in *Appendix C*.

To download BioCyc databases, you can select the specific databases and download them one by one from the BioCyc website, or they can all be downloaded together. Refer to the website for download instructions.

The data files from which BioCyc data are imported into PathBlazer are provided in a defined format as specified by BioCyc: (http://brg.ai.sri.com/ptools/flatfile-format.html). PathBlazer 2.0 was specifically designed to import only those files defined by BioCyc in their Flat File Format.

BioCyc Source Files[1]

Import of BioCyc data will be described using the Ecoo157Cyc file--the database that contains information about *E. coli* H:0157.

Download either the **Ecoo157Cyc-flatfiles.zip** or **Ecoo157Cyc-flatfiles.tar.z**, according to your preference, and unzip it. In your root folder, the following source file(s) will display:

bindrxns.dat *

classes.dat (this file is not used for BioCyc import)

compounds.dat

dnabindsites.dat *

ecobase.ocelot (this file is not used for BioCyc import)

enzrxns.dat

enzymes.col

genes.col

genes.dat

pathways.col*

pathways.dat

promoters.dat*

protcplxs.col

proteins.dat

protseq.fasta (this file is not used for BioCyc import)

pubs.dat

reactions.dat

regulons.dat*

terminators.dat*

transporters.col

transunits.dat*

BioCyc Import Logic

Pathblazer component information is assembled from the following files:

compounds.dat; dnabindsites.dat*; enzymes.col; genes.col; genes.dat; promoters.dat*; protcplxs.col; proteins.dat; regulons.dat*; terminators.dat*; transporters.col; transunits.dat

Reaction information is gathered from findrxns.dat and reactions.dat.

Pathways information is loaded from pathways.col and pathways.dat.

1. The data files from which BioCyc data are imported into PathBlazer are provided in a defined format as specified by BioCyc: (http://brg.ai.sri.com/ptools/flatfile-format.html). In this section are examples of the data from the public dataset to show the files and fields used to populate Vector PathBlazer during import. PathBlazer 2.0 was specifically designed to import only those files defined by BioCyc in their Flat File Format. Files marked with asterisks are used to import EcoCyc and MetaCyc databases and are not described in this manual.

BioCyc Component Files

File compounds.dat

The compound.data file is a collection of organic and inorganic substances that cannot be classified as nucleic acid or enzyme. These components are identifieid by their UNIQUE-ID. Compounds have information about their type, commonly used synonyms, atomic charges, chemical formulae, links to external databases, etc.

The following is an example of components extracted from the Ecoo157Cyc compounds.dat file. The values shown in bold are parsed by PathBlazer 2.0. The chemical formula is reconstructed. For a component below, the formula will be C12H17N401S1. HTML specific tags like _{and} are removed from names. The synonym of the component below is Vitamin B1.

UNIQUE ID	**THIAMINE**
TYPES	Vitamins
COMMON NAME	**thiamine**
ATOM-CHARGES	**(**9 1**)**
CHEMICAL FORMULA	**(C 12)**
CHEMICAL FORMULA	**(H 17)**
CHEMICAL FORMULA	**(N 4)**
CHEMICAL FORMULA	**(O 1)**
CHEMICAL FORMULA	**(S 1)**
DBLINKS	**(CAS “59-43-8”))**
MOLECULAR WEIGHT	**265.352**
SMILES	c1(c(cnc(C)n1)C[n+1]2(c(C)c(sc2)CCO))(N)
SYNONYMS	**thiamin**
SYNONYMS	**vitamin B₁**

File enzymes.col

This file contains information about enzymes. This is a tabular format file.

A one line excerpt for glucokinase is given below. A component in PathBlazer named **ENZRXN7E-124** with the synonym **glucokinase** and type **Enzyme** is created. The annotations store information that this enzyme catalyzes the reaction

beta-D-Glucose + ATP = beta-D-glucose 6-phosphate + ADP.

This enzyme is found in TREDEGLOW-PWY and acts as monomer.

ENZRXN7E-124 glucokinase β-D-glucose + ATP = β-D-glucose 6-phosphate + ADP phosphate + ADP TREDEGLOW-PWY 1*GLK-MONOMER

File genes.col and genes.dat

These two files contain description of genes. Components in PathBlazer are created. The name of an entry is augmented with "/gene/". These components have class DNA > Chromosome > Gene.

The following is an example of a genes.col file entry.

ZNTA	**zntA zinc-transporting ATPase ECOLIO157 4392922 4395120 UNCLASSIFIED**
…	

The corresponding entry in the genes.dat file:

UNIQUE ID	**ZNTA**
TYPES	**Unclassified-Genes**
COMMON NAME	**zntA**
CENTISOME-POSITION	79.46035
COMMENT	**Residues 1 to 732 of 732 are 98.90 pct identical to residues 1 to 732 of 732 from Escherichia coli K-12 Strain MG1655: B3469**
COMPONENT-OF	ECOLIO157
LEFT-END-POSITION	**4392922**
PRODUCT	**ZNTA-MONOMER**
RIGHT-END-POSITION	**4395120**
TRANSCRIPTION-DIRECTION	**+**

The name of the component is taken from the COMMON-NAME field. A UNIQUE-ID and first column in the genes.col file will be added to the synonyms list. The content of PRODUCT filed will be added to the description. The start and end of the gene described in LEFT-END-POSITION and RIGHT-END-POSITION will be entered into the description.

File protcplxs.dat

This file contains information about protein complexes. It is in tabular format. They are stored as components in PathBlazer database.

An entry from this file:

CPLX7E-9	**glycine tRNA synthetase glyQ glyS GLYQ GLYS** 2*GLYQ-MONOMER,2*GLYS-MONOMER

The component will be named **glycine tRNA synthetase**. The gene names of proteins forming this complex, **glyQ** and **glyS** will be stored as crosslinks to EcoCyc genes information and in the component description. Subunits **GLYQ** and **GLYS** will be crosslinked to protein information and stored in the description as well.

File proteins.dat

This file contains information about proteins/polypeptides, which do not have EC classification. Components classified as proteins are created in the PathBlazer database.

The following is an entry from the proteins.dat file. Parts which are extracted by PathBlazer are shown in bold. If MODIFIED-FORM or UNMODIFIED-FORM is present, the modification/ unmodification reaction

RED-THIOREDOXIN-MONOMER->OX-THIOREDOXIN-MONOMER

is created and stored as a separate reaction object.

UNIQUE ID	**RED-THIOREDOXIN-MONOMER**
TYPES	red-thioredoxin
COMMON NAME	**thioredoxin 1**
COMMENT	**enzyme; Biosynthesis of cofactors, carriers: Thioredoxin, glutaredoxin, glutathione**
GENE	**TRXA**
LOCATIONS	**INNER-MEMBRANE**
MODIFIED FORM	**OX-THIOREDOXIN-MONOMER**
SPECIES	**E. coli**
SYNONYM	**reduced thioredoxin**
SYNONYM	**TrxA**
SYNONYM	**thioredoxin(SH)₂**

File transporters.col

This file contains information about transporters. The file is in tabular format.

An entry:

Z3799-MONOMER putative ATP synthase beta subunit H+[cytoplasm] + H2O + ATP =H+[periplasm] + phosphate + ADP 1*Z3799-MONOMER

A component named putative **ATP synthase beta subunit** with the synonym **Z3799-MONOMER** will be created. The reaction equation **H+[cytoplasm] + H2O + ATP =H+[periplasm] + phosphate + ADP** and the subunit composition will be entered into the description field.

BioCyc Reaction Files

Reaction files describe reactions. They are parsed into reaction objects in the PathBlazer database. Some reactions described as a simple reaction in BioCyc files will be parsed into more then one reaction in PathBlazer. For example, an enzymatic reaction with an enzyme being activated by some other compound will result in two reactions in database, one describing an enzymatic reaction by itself, the other a reaction of enzyme activation. The parsing of reactions starts from reactions.dat. Later, non-redundant information is added from files bindrxns.dat and enzrxns.dat.

File reactions.dat

This file contains general information about reactions. The reaction is constructed from data stored in this file as well as from references made to other files of BioCyc.

The following is an entry from the reactions.dat file.

UNIQUE ID	**R81-RXN**
TYPES	EC-2.7.1
COMMON NAME	**Hexokinase**
EC NUMBER	**2.7.1.1**
IN-PATHWAY	**ANAGLYCOLYSIS-PWY**
IN-PATHWAY	**P122-PWY**
LEFT	**GLC**
LEFT	**ATP**
OFFICIAL-EC?	NIL
RIGHT	**GLC-6-P**
RIGHT	**ADP**
SYNONYM	Hexokinase type IV
SYNONYM	Glucokinase
…	

The following reaction will be reconstructed:

beta-D-glucose + ATP => glucose-6-phosphate+ADP catalyzed by Hexokinase.

GLC corresponds to **beta-D-glucose** in the compound.dat file, **ATP** to **ATP**, **GLC-6-P** to **glucose-6-phosphate**, etc. The enzyme **Hexokinase** is described in the enzymes.col file. The reaction will be named **R81-RXN**.

File enzrxns.dat

This file contains information about enzymatic reactions. The reaction is constructed from data stored in this file as well as from references made to other files of BioCyc. Only information which is different from that stored in the reactions.dat file is loaded from the enzrxns.dat file.

An entry from this file is shown below.

UNIQUE ID	ENZRXN7E-124
TYPES	Enzymatic-Reactions
COMMON NAME	glucokinase
BASIS-FOR-ASSIGNMENT	MANUAL
ENZYME	GLK-MONOMER
REACTION	GLUCOKIN-RXN
…	

BioCyc Pathways File

BioCyc databases store information about pathways. Pathways are stored as pathway objects in the PathBlazer database.

File pathways.dat

One pathway from this file is shown below.

UNIQUE ID	P122-PWY
TYPES	Fermentation
COMMON NAME	**heterofermentative lactate fermentation**
PREDECESSORS	(**ALCOHOL-DEHYDROG-RXN ACETALD-DEHYDROG-RXN**)
PREDECESSORS	**(ACETALD-DEHYDROG-RXN PHOSACETYLTRANS-RXN)**
PREDECESSORS	**(PHOSACETYLTRANS-RXN PHOSPHOKETOLASE-RXN)**
PREDECESSORS	**(DLACTDEHYDROGNAD-RXN PEPDEPHOS-RXN)**
PREDECESSORS	**(PEPDEPHOS-RXN 2PGADEHYDRAT-RXN)**
PREDECESSORS	**(2PGADEHYDRAT-RXN 3PGAREARR-RXN)**
PREDECESSORS	**(3PGAREARR-RXN PHOSGLYPHOS-RXN)**
PREDECESSORS	**(PHOSGLYPHOS-RXN GAPOXNPHOSPHN-RXN)**
PREDECESSORS	**(PHOSGLYPHOS-RXN 1.2.1.13-RXN)**
PREDECESSORS	**(GAPOXNPHOSPHN-RXN PHOSPHOKETOLASE-RXN)**
PREDECESSORS	**(1.2.1.13-RXN PHOSPHOKETOLASE-RXN)**
PREDECESSORS	**(PHOSPHOKETOLASE-RXN RIBULP3EPIM-RXN)**
PREDECESSORS	**(RIBULP3EPIM-RXN 6PGLUCONDEHYDROG-RXN)**
PREDECESSORS	**(6PGLUCONDEHYDROG-RXN R84-RXN)**
PREDECESSORS	**(R84-RXN R81-RXN)**
REACTION-LIST	ALCOHOL-DEHYDROG-RXN
REACTION-LIST	ACETALD-DEHYDROG-RXN
REACTION-LIST	PHOSACETYLTRANS-RXN
REACTION-LIST	DLACTDEHYDROGNAD-RXN
REACTION-LIST	PEPDEPHOS-RXN
REACTION-LIST	2PGADEHYDRAT-RXN
REACTION-LIST	3PGAREARR-RXN
REACTION-LIST	PHOSGLYPHOS-RXN
REACTION-LIST	1.2.1.13-RXN
REACTION-LIST	GAPOXNPHOSPHN-RXN
REACTION-LIST	PHOSPHOKETOLASE-RXN
REACTION-LIST	RIBULP3EPIM-RXN
REACTION-LIST	6PGLUCONDEHYDROG-RXN
REACTION-LIST	R84-RXN
REACTION-LIST	R81-RXN
SPECIES	HPY
…	

The pathway is stored under name 'heterofermentative lactate fermentation' and is assembled from the highlighted reactions.

Instructions for Importing BioCyc Data

You must have downloaded the data files described in *BioCyc Source Files* on page 86 to your local file system. The databases from BioCyc collection can be imported separately, or as a group. You can import data either into the default PathBlazer database or into a new separate database you create before the data import. To create a database, see *Creating a New Database* on page 10.

Use the following steps to import BioCyc data into the Vector PathBlazer database.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. From an open PathBlazer window, select **File > Import.**
3. In the PathBlazer Import dialog box, the first screen of the Import Wizard, select **Import BioCyc Data**. The **Description** box reflects the type of data chosen for import. Click **Next**.

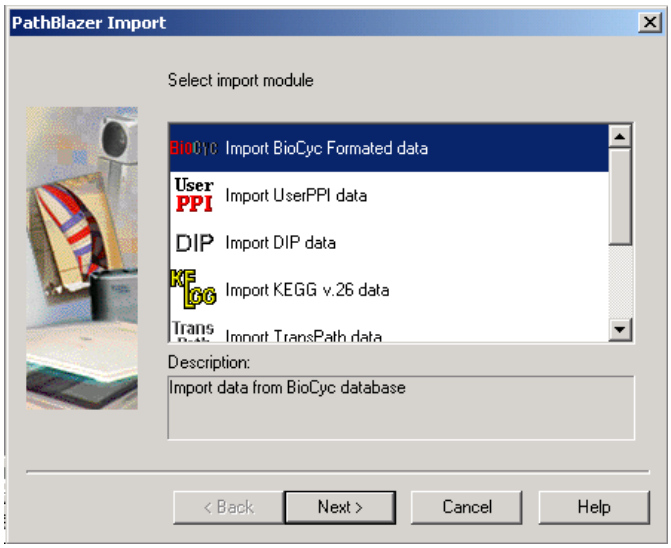


Figure 4.11 BioCyc file selected for import

4. In Screen 2 of the Import Wizard, enter the name of the organism whose genes you are going to import (*Figure 4.11*).

Important: This organism will be applied to all entries from this database with the quantifier KNOWN IN unless specific information in a specific entry contradicts it. For example, if you import the AgroCyc database, you might want to enter *Agrobacterium tumifaciens*. This organism name will be applied to all entries taken from the AgroCyc database. The organism field can be left empty if, for example, you import the entire BioCyc database or if for some reason you do not want to specify an organism.

5. Select the root folder storing the multiple BioCyc files by clicking the **Browse** button. Locate the corresponding file in the **Browse for Folder** dialog box and click **OK**. The complete path to the root folder file displays in the **Root Folder** field (*Figure 4.11*).

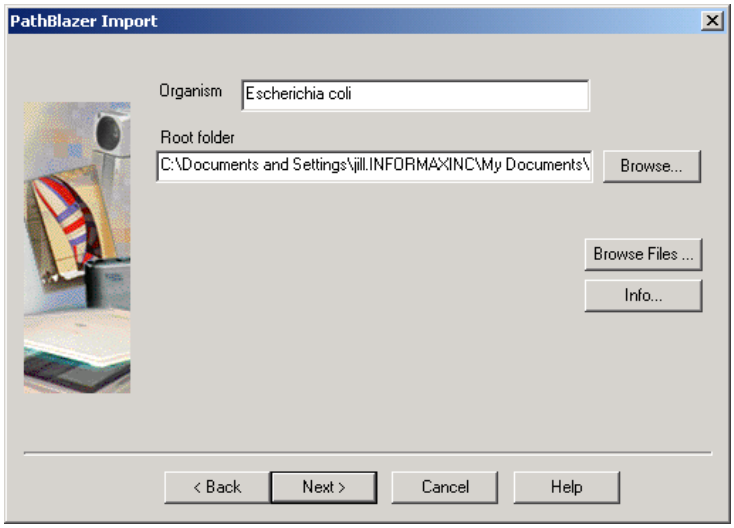


Figure 4.12 BioCyc Import dialog box for selecting the root folder

6. In the Merge Options dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information. Click **Next** to continue.

   The data loads while a monitor displays, allowing you to follow the import progress. An import log summarizing import results displays when import has been successfully completed.

7. Click **Close**.
8. To import the data, click **Next**. The **Load BIND** dialog box opens and displays the progress of the import. To stop the import, click **Cancel**.
9. A message displays when import is successfully completed. Click **Close**.

BioCyc data includes components, reactions or pathways, so these objects will be distributed into all of these PathBlazer folders. (BioCyc is the only publicly available database with pathway objects.)

Importing TransPath Data

TransPath is comprised of molecules that participate in signal transduction and their reactions, thus creating a complex network of interconnected signaling components. TransPath focuses on signaling cascades that aim at transcription factors and thus alter the gene expression profile of a given cell, helping to bridge the gap between extra cellular signal molecules (such as hormones, cytokines etc.) and the genes responding to these triggers.[1]

A complete description of the contents of TransPath as well as licensing information is available at http://www.biobase.de/pages/products/transpath.html. Reference and marketing information is also available in *Appendix C*.

TransPath Source Files

Upon downloading the TransPath database, save the source files in the same folder. Two files, molecule.xml and reaction.xml are essential for import. Files gene.xml, annotate.xml, loca-

1. http://transpath.gbf.de/

tion.xml, reference.xml and hyperlinks.xml are non-essential for PathBlazer import. They contain some auxiliary information.

TransPath starts from the parsing of molecule.xml, which contains information about molecules. They are stored as components in the PathBlazer database. During the import of reactions.xml, these components are connected into reactions. Some additional reaction annotations are also extracted from this file.

File molecule.xml

This file contains information about components. Each component has a unique id specified by <Molecule id=…> tag. The name of a component is defined by <name> tag, and synonyms by <synonyms> tags.

An excerpt from the molecule.xml file describing glucose is shown below. The fields which are extracted are highlighted. The <comments> and <references> tags are used to define crosslinks to other objects inside TransPath (e.g. to reactions) as well as to objects in external databases.

```
<Molecule id="MO000021249">
<!-- Copyright (c) Biobase GmbH -->
 <creator>mkl</creator>
 <updator>mkl</updator>
 <type>other</type>
 <name>glucose</name>
 <synonyms>Glc</synonyms>
 <comments>
<item type="Annotate" xlink:type="simple" xlink:href="annotate.xml#ID
(AN000031352)" xlink:show="new" xlink:actuate="onRequest">AN000031352</item>
….
 </comments>
<references>
…
 </references>
……
</Molecule>
```

File reaction.xml

This file stores information about reactions and references to components in molecule.xml.

An excerpt describing one reaction is shown below. The reaction will have the unique id XN000000001. Its formula will be GTP + Ras:GDP -GEF-> Ras:GTP + GDP. GEF plays an enzymatic role in this reaction. Tags <reactants>, <products> and <enzymes> contain references to respective molecules in the molecule.xml file.

```
<Reaction id="XN000000001">
<!-- Copyright (c) Biobase GmbH -->
 <creator>frs</creator>
 <updator>frs</updator>
 <type>mechanistic</type>
 <name>GTP + Ras:GDP -GEF-&gt; Ras:GTP + GDP</name>
 <effect>exchange</effect>
 <reversible>false</reversible>
 <references>
….
```

```
<reactants>
<item type="Molecule" xlink:type="simple" xlink:href="molecule.xml#ID
(MO000000005)" xlink:show="new" xlink:actuate="onRequest">MO000000005</item>
<item type="Molecule" xlink:type="simple" xlink:href="molecule.xml#ID
(MO000000007)" xlink:show="new" xlink:actuate="onRequest">MO000000007</item>
</reactants>
<produces>
<item type="Molecule" xlink:type="simple" xlink:href="molecule.xml#ID
(MO000000004)" xlink:show="new" xlink:actuate="onRequest">MO000000004</item>
<item type="Molecule" xlink:type="simple" xlink:href="molecule.xml#ID
(MO000000006)" xlink:show="new" xlink:actuate="onRequest">MO000000006</item>
</produces>
<enzyme>
<item type="Molecule" xlink:type="simple" xlink:href="molecule.xml#ID
(MO000000024)" xlink:show="new" xlink:actuate="onRequest">MO000000024</item>
</enzyme>
</Reaction>
```

TransPath Auxiliary files

gene.xml

This file contains information about genes. Only information not contained in molecule.xml is extracted from this file. Some additional links are stored as crosslinks to external databases.

annotate.xml

This file contains additional annotations about function, structure, kinetics, mechanism, methods, etc. for components and reactions. Some additional information is entered into external crosslinks.

reference.xml

References to scientific publications are extracted from this file. They are stored inthe description field.

location.xml

This file contains information about subcellular location. This information is stored in component and reaction location fields.

hyperlinks.xml

This file contains links to external databases for molecules and reactions.

Custom Dictionaries

There are two TransPath custom dictionaries created in Vector PathBlazer: **classDict** and **organDict**. These dictionaries are supplied with PathBlazer, and they are text tab-delimited files. The file **classDict** contains the dictionary that translates classes of molecules as they are defined in TransPath into an internal PathBlazer classification. File **organDict** translates names of organisms according to TransPath usage into PathBlazer names. There is no need for the ordinary user to modify or amend these files, but an advanced user may want to change the classification mapping.

Instructions for Importing TransPath Data

You can import data either into the default PathBlazer database or into a new separate database you create before the data import. To create a database, see *Creating a New Database* on page 10. You must also have downloaded the data file described in *TransPath Source Files* on page 93 to your local file system.

Use the following steps to import TransPath data into the Vector PathBlazer database.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. From an open PathBlazer window, select **File > Import**. The PathBlazer Import tool opens, displaying the various import options (*Figure 4.13*).

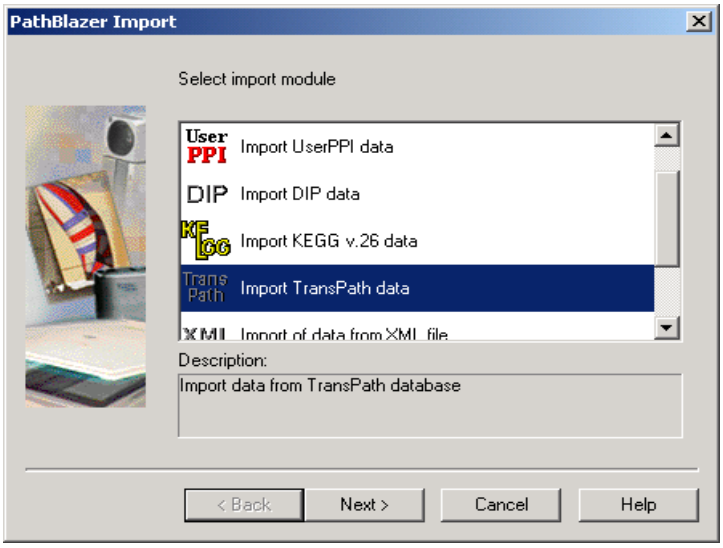


Figure 4.13 TransPath file selected for import

3. Choose **Import TransPath Data**. The **Description** box reflects the type of data chosen for import. Click **Next**.
4. In Screen 2 of the Import Wizard, in the Root Folder field, locate the root folder storing the multiple TransPath files by clicking the **Browse** button (*Figure 4.13*). Select the correct folder in the **Browse for Folder** dialog box and click **OK**. The complete path to the root folder file displays in the **Root Folder** field.

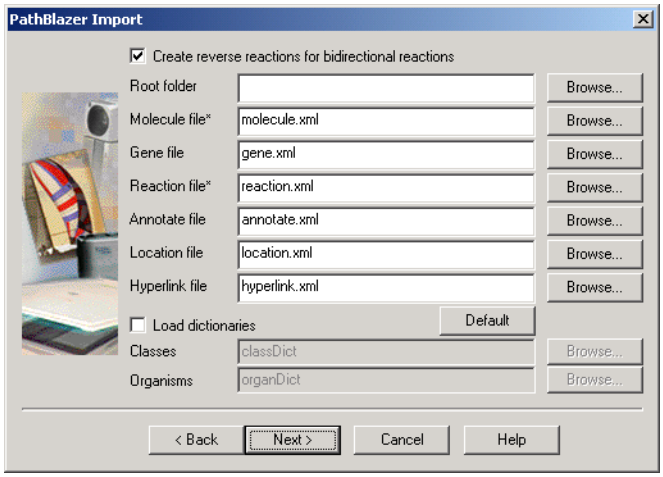


Figure 4.14 TransPath Import dialog box for selecting the root folder and source files

The other fields in this dialog box display the .xml file names for the TransPath data. These files are found in the root folder, and you shouldn't have to locate them unless they are stored outside that folder.

Note: Only files labeled in the import window by asterisks are absolutely required for successful import.

- *Optional*: Check the **Create Reverse Reactions for Bidirectional Reactions** checkbox to execute that option.
- *Optional:* Check the **Load Dictionaries** checkbox to use the custom dictionaries. You will need to browse for the **classDict** and **organDict** files. For more information about dictionaries, see *Custom Dictionaries* on page 95.

5. In the Merge Options dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information. Click **Next** to continue.

   The data loads while a monitor displays, allowing you to follow the import progress. An import log summarizing import results displays when import has been successfully completed.

6. Click **Close**.

Importing DIP Data

DIP (Database of Interacting Proteins) is a database that documents experimentally determined protein-protein interactions. This database is intended to provide data for extracting information about protein interactions and interaction networks in biological processes. [1]

A complete description of the contents of the DIP database as well as licensing information is available at http://dip.doe-mbi.ucla.edu/hold/. Reference and licensing information is also available in *Appendix C*.

DIP Source Files and Import Logic

The file *dipYYYMMDD.xin* is used to load DIP data into the Vector PathBlazer database where *YYYYMMDD* is the date of a database release (for example: dip20020616.xin). Download this file from http://dip.doe-mbi.ucla.edu/dip/Download.cgi.

The file consists of two parts: components and reactions. A component object is created for each component listed in the file. The following is a partial example of the part of the file that contains component information. Values of attributes or elements that are directly parsed are in bold.

XML Source:

```
<node uid="DIP:3N" id="3" name="RA52_YEAST" class="protein">
…
<att name="organism">
  <val>Saccharomyces cerevisiae (budding yeast)</val>
 </att>
</node>
```

1. http://dip.doe-mbi.ucla.edu/hold/

The following table shows the XML tag that is parsed from the component part, a description of its value, and where the value displays in the program (Table *4.5*).

XML tag	***Description***	***Annotation in Vector PathBlazer***
<node name=> </node>	Recommended name	Component Name **Note:** Components named "UNDEFINED", "UNKNOWN", "-", "Homo sapiens" or an empty value are skipped.
<node uid> </node>	Alternate name	Component Synonym Component Crosslink
<node class=> </node>	Molecule type. All molecules in the DIP database are proteins.	Component Class
<att name="organism">	Species	Component Organism
<att name="descr">	Description of the components	Component Synonyms
<feature name= >	Links to other databases	Component CrossLinks

Table 4.5 XML tags that are imported for DIP components

A list of pre-defined URLs are automatically setup for DIP components and are placed in the CrossLinks annotation field. These are listed in *Pre-Defined URLs* on page 107.

A reaction object is created for each reaction listed in the file. The following is a partial example of the part of the file that contains reaction information. Values of attributes or elements that are directly parsed are in bold. Components are linked to a reaction using the values in the **from** and **to** fields in the reaction part of the file, which correspond to the value in the **id** field for a component in the component part.

```
<edge uid="DIP:17861E" id="17692" from="4692" to="1603" class="inter">
 <feature name="DIP:21686X" class="exp:s">
  <src>PMID:12011112</src>
  <val>Affinity column</val>
 </feature>
 <att name="class">
  <val>core</val>
 </att>
</edge>
```

The following table shows the XML tag that is parsed from the reaction part, a description of its value, and where the value displays in the program (Table *4.6*).

XML tag	*Description*	*Annotation in Vector PathBlazer*
<edge uid=> </edge>	Primary name	Reaction Name Example: DIP:17861E Reaction Crosslink **Note:** Reactions including components named “UNDEFINED”, “UNKNOWN”, “-”, “Homo sapiens” or an empty value are skipped.
<src></src>	PubMed ID	Reaction Crosslinks

Table 4.6 XML tags that are imported for DIP components

A list of pre-defined URLs that are automatically setup for DIP reactions and are placed in the CrossLinks annotation field. These are listed in *Pre-Defined URLs* on page 107.

Instructions for Importing DIP

You can import data either into the default PathBlazer database or into a new separate database you create before the data import. To create a database, see *Creating a New Database* on page 10. You must also have downloaded the data file described in *DIP Source Files and Import Logic* on page 97 to your local file system.

Use the following steps to import DIP data into the Vector PathBlazer database.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. From an open PathBlazer window, select **File > Import**. The PathBlazer Import tool opens, displaying the various import options (*Figure 4.15*).

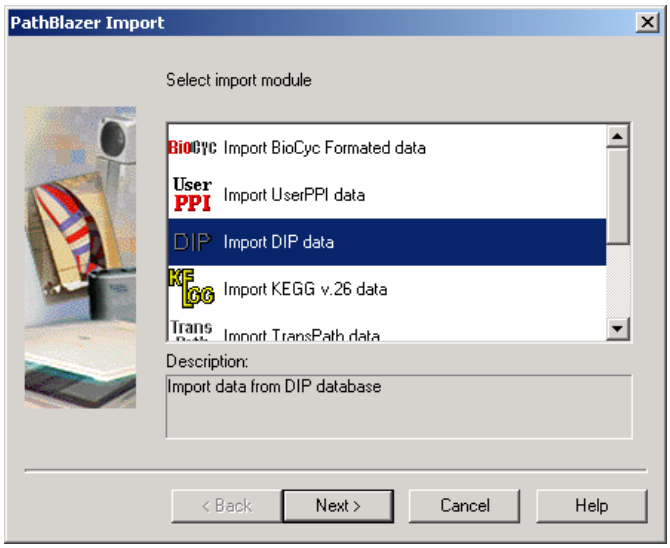


Figure 4.15 DIP file selected for import

3. Shoose **Import DIP Data**. The **Description** box reflects the type of data chosen for import. Click **Next**.

4. In Screen 2 of the Import Wizard, select the DIP.xml file for import by clicking the **Browse** button, locating the corresponding file in the **Open** dialog box, and clicking **Open**. The complete path to the file displays in the **Select source file** field (*Figure 4.10*). Click Next to continue.

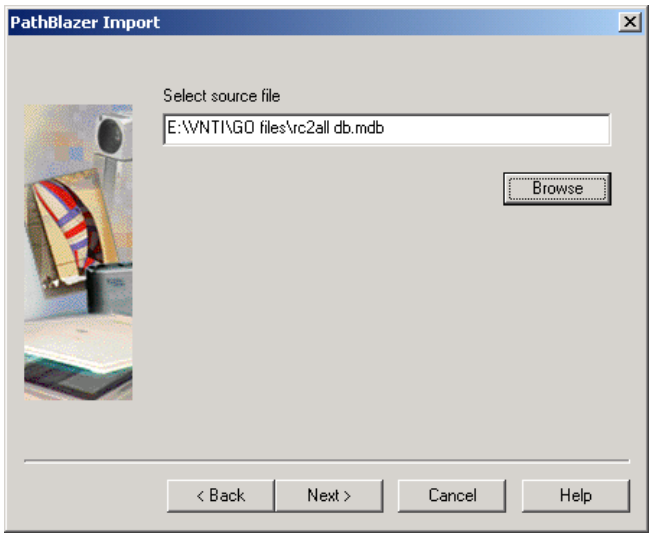


Figure 4.16 Dip Import dialog box for selecting the DIP source file

5. In the Merge Options dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information. Click **Next** to continue.

   The data loads while a monitor displays, allowing you to follow the import process. An import log summarizing import results displays when the import has been successfully complete. displays the progress of the import.

6. To stop the import, click **Cancel**. Click **Close**.
7. Once imported, verify the import process by choosing an example of a KEGG reaction in the Graphics window.

Similar to BIND, DIP data contains only information about interactions between two components (that is, proteins). There are no predicted products and each DIP reaction is represented as a protein-protein interaction.

Importing PPI Data

When a sequence of a protein is known, clues to the correlation of the protein sequence and its structure to its functionality begin to unfold. Domains, usually the functional regions of a protein molecule, can interact with a wide range of cellular objects including domains on other proteins. The interactions of proteins with each other and the strength of the interactions helps scientists to visualize and correlate protein pathway data and chart protein pathways within cells.

PathBlazer allows you to view a network of proteins linked by their domains to ligand interactions. You can display, analyze and manipulate a graphical representation of a PPI (protein-protein interaction) network.

PPI data import is a simple process in PathBlazer. Prepare the data in a 3-column tab-delimited file, with a column for each protein A, protein B, and the strength of the interaction (affinity).

Instructions for Importing User PPI Data

You can import data either into the default PathBlazer database or into a new separate database you create before the data import. To create a database, see *Creating a New Database* on page 10.

Use the following steps to import PPI data into the Vector PathBlazer database.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. From an open PathBlazer window, select **File > Import.** The PathBlazer Import tool opens, displaying the various import options (*Figure 4.17*).

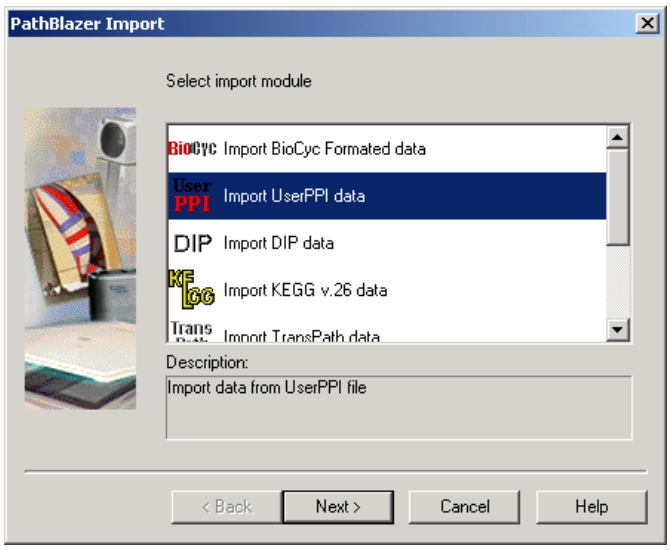


Figure 4.17 User PPI file selected for import

3. In Screen 2 of the Import Wizard, select the User PPI file for import by clicking the **Browse** button. Locate the corresponding file in the **Open** dialog box, and click **Open**. The complete path to the file displays in the **Select source file** field (*Figure 4.18*). Click **Next** to continue.

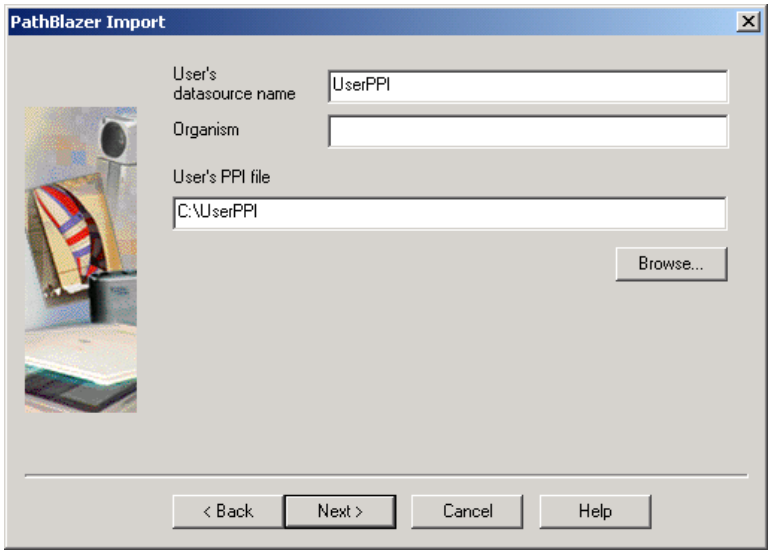


Figure 4.18 PPI Import dialog box for selecting source file

4. In the Merge Options dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information. Click **Next** to continue.

   The data loads while a monitor displays, allowing you to follow the import process. An import log summarizing import results displays when the import has been successfully complete.
5. To stop the import, click **Cancel**. Click **Close**.

When you have completed the import, the components and reactions relating to the imported file display in the Explorer List Pane, with the PPI datasource listed in the Database Source column.

Similar to DIP and BIND, PPI data contains only information about interactions between two components (that is, proteins). There are no predicted products and each PPI reaction is represented as a protein-protein interaction.

Importing Proprietary Data

Proprietary data in the form of components, reactions, and pathways can be imported into the Vector PathBlazer database by formatting the data in an XML file according to the DTD (Document Type Definition) for Vector PathBlazer. The format of the XML file for proprietary data is the same XML format into which public data is automatically converted by the program for import into the database.

The complete DTD for formatting Vector PathBlazer XML files is provided in *Appendix B*. An example XML file is provided in the following sections that you can use to format a proprietary file. The XML file is made up of three main parts: a list of substances (that is, components), a list of the reaction or list of pathways.

Defining Components

The first part of the file contains a list of substances (that is, components) and the attributes of each component. For each component described between the **<substance>** attribute, a component object is created in the database for which the unique ID is included in the **<substance ID>** element. The value of <substance ID> is referenced by any reactions in which the component is included. Each of the elements between the attribute **<substance>** describe annotations of the component.

```
...
  <list_of_substances>
    <substance ID="Phosphopyruvate hydratase" DB="KEGG" Disease="" Source=”" Description="Also acts on 3-phospho-D-erythronate. Ki of phosphonoacetohydroxamate is 15 picoM as the trianion with saturation Mg++ ion (Biochemistry, 1984, 23, 2779). Crystal structure of the inhibitor complex (Biochemistry, 1994, 33, 6295-6300).">
      <synonyms>
        <name>2-Phospho-D-glucerate hydro-lyase</name>
        <name>2-Phosphoglycerate dehydratase</name>
        <name>EC 4.2.1.11</name>
        <name>Enolase</name>
        <name>Phosphopyruvate hydratase</name>
      </synonyms>
      <type>Protein|Enzyme|EC EC 4.2.1.11</type>
      <list_of_origin_accesses>
        <origin_access Text="">
            <database>AAE</database>
            <access>aq_484(eno)</access>
        </origin_access>
```

```
      <origin_access Text="">
        <url>http://www.rcsb.org/pdb/cgi/explore.cgi?pdbId=7ENL</url>
      </origin_access>
    </list_of_origin_accesses>
    <organisms>
      <organism Class="0" Name="Aeropyrum pernix" log_op="154137832" />
      <organism Class="0" Name="Agrobacterium tumefaciens" log_op="196791" />
      <organism Class="0" Name="Anabaena sp." log_op="154138064" />
    </organisms>
  </substance>
...
```

Defining Reactions

The second part of the file contains a list of interactions (that is, reactions) and the attributes of each reaction. For each reaction described between the **<reaction>** attribute, a reaction object is created in the database with the unique ID that is included in the **<reaction ID>** element. Each of the attributes between the **<reaction>** attribute describe annotation of the reaction. Components in a reaction are included in the **<agent ID>** element. If a reaction is directional, the **<role>** attribute contains **educt** to indicate the component is a substrate of the reaction or **product** to indicate the component is a product. The actual reference to the component in the component part of the file is included in the **<substance ref>** element. Each connector in the reaction is defined in the **<conf_arc>** attribute, which describes the “from” component to its appropriate reaction.

```
...
<list_of_interactions>
  <reaction ID="Gly 1" DB="" Descr="" Type="unknown">
    <BioNet ID="Gly 1">
      <list_of_agents>
        <agent ID="3-(ADP)-2-phosphoglycerate">
          <role>educt</role>
          <substance ref="3-(ADP)-2-phosphoglycerate" />
        </agent>
        <agent ID="Phosphoenolpyruvate">
          <role>product</role>
          <substance ref="Phosphoenolpyruvate" />
        </agent>
        <agent ID="H2O">
          <role>product</role>
          <substance ref="H2O" />
        </agent>
        <agent ID="Phosphopyruvate hydratase">
```

```
            <role>catalyzing_agent</role>
            <substance ref="Phosphopyruvate hydratase" />
          </agent>
        </list_of_agents>
        <list_of_actions>
          <action ID="Gly 1">
            <reaction ref="Gly 1" />
          </action>
        </list_of_actions>
        <list_of_arcs>
          <conf_arc from="3-(ADP)-2-phosphoglycerate" to="Gly 1"
          TransitionProbability="0">
            <bidirect>No</bidirect>
            <type>ordinary</type>
            <weight>1</weight>
          </conf_arc>
          <conf_arc from="Gly 1" to="Phosphoenolpyruvate"
          TransitionProbability="0">
            <bidirect>No</bidirect>
            <type>ordinary</type>
            <weight>1</weight>
          </conf_arc>
          ...
        </list_of_arcs>
      </BioNet>
    </reaction>
  ...
```

Defining Pathways

The third part of the file contains a list of pathways. Each pathway contains a list of components and reactions. For each pathway described between the **<BioNet ID>** attribute, a pathway object is created in the database with the unique ID that is included in that attribute. The **<list_of_agents>** attribute describes each component in the pathway. The name of the actual component is determined from the **<substance ref>** element, whose value is matched to the **<substance ID>** attribute in the component part of the file. The **<list_of_actions>** attribute describes each reaction in the file. The name of the actual reaction is determined by matching the value of **<reaction ref/>** to **<reaction ID>** in the reaction part of the file. Finally, each connector in the reaction is defined in the **<conf_arc>** attribute, which describes the "from" component to its appropriate reaction.

```
...
  <pathway ID="glycolysis" DB="" Disease="" Desrc="" InternalID="1038">
    <BioNet ID="glycolysis">
```

```
<list_of_agents>
  <agent ID="AGENT0001" >
    <substance ref="Phosphopyruvate hydratase" />
  </agent>
  <agent ID="AGENT0002">
    <substance ref="ADP" />
  </agent>
  <agent ID="AGENT0003">
    <substance ref="D-Glucose" />
  </agent>
<list_of_agents>
<list_of_actions>
  <action ID="INTERACTION0001" Type="unknown">
    <reaction ref="Gly 1" />
  </action>
  <action ID="INTERACTION0002" Type="unknown">
    <reaction ref="Gly 2" />
  </action>
</list_of_actions>
<conf_arc from="AGENT0003" to="INTERACTION0005" TransitionProbability="0">
  <bidirect>No</bidirect>
  <type>ordinary</type>
  <weight>1</weight>
</conf_arc>
<conf_arc from="INTERACTION0005" to="AGENT0025" TransitionProbability="0">
  <bidirect>No</bidirect>
  <type>ordinary</type>
  <weight>1</weight>
</conf_arc>
</BioNet>
</pathway>
...
</list_of_pathways>
```

Instructions for Importing Proprietary Data

You can import data either into the default PathBlazer database or into a new separate database you create before the data import. To create a database, see *Creating a New Database* on page 10. You must also have formatted proprietary data according in Vector PathBlazer XML file format described above. The complete DTD is included in *Appendix B*.

Use the following steps to import proprietary data into the Vector PathBlazer database.

1. Backup the database into which the data will be imported. For instructions, see *Backing Up the Database* on page 11.
2. From an open PathBlazer window, select **File > Import.** The PathBlazer Import tool opens, displaying the various import options (*Figure 4.19*).

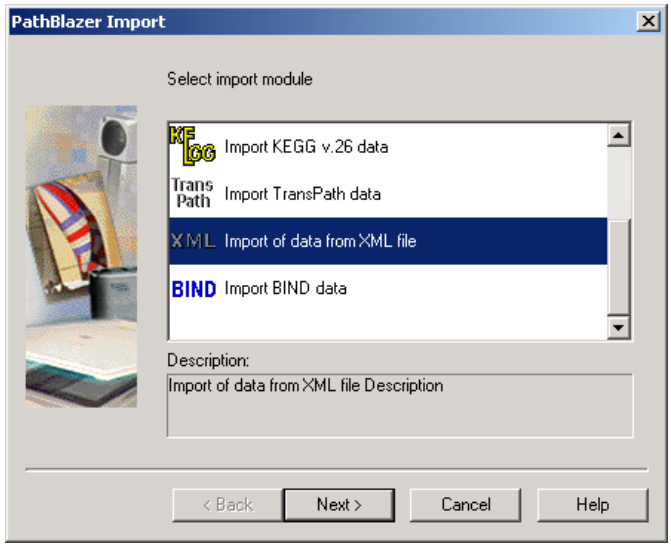


Figure 4.19 XML file selected for import

3. In the **Select Import Module** box, choose **Import of data from XML file**. The **Description** box reflects the type of data chosen for import. Click **Next**.
4. In Screen 2 of the Import Wizard, select the .xml file for import by clicking the **Browse** button, locating the corresponding file in the **Open** dialog box, and clicking **Open**. The complete path to the file displays in the **Select source file** field (*Figure 4.20*).

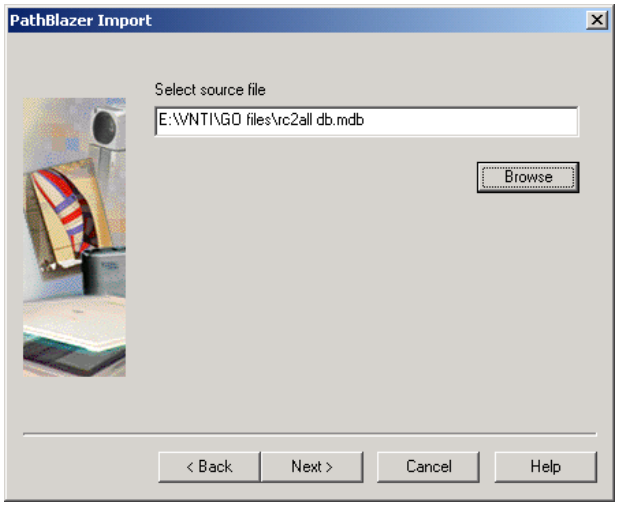


Figure 4.20 Proprietary Import dialog box for selecting XML source file

5. In the Merge Options dialog box, select the options appropriate for merging the data. See *Merge Option Dialog Box* on page 67 for more information. Click **Next** to continue.

   The data loads while a monitor displays, allowing you to follow the import process. An import log summarizing import results displays when the import has been successfully complete.
6. To stop the import, click **Cancel**. Click **Close**.

Pre-Defined URLs

Some pre-defined URLs are automatically associated with imported entries. The associated link depends on the type and source of entry. For example, a BIND component is associated with a link to the BIND database by the entry value of the component. Pre-defined URLs are listed in Table *4.7*.with the name that is displayed for each link in the program. Examples are shown with an actual entry value but this value depends on the entry number of each component and reaction.

Description	*URL*	*Display Name*
Bind Component	http://bind.ca/cgi-bin/bind/dataget?get=tindex&text_query=%s&iid_cb=4&mcid_cb=16&pid_cb=8&npp=20&submit=Submit	BIND Protein Link
Bind Reaction Link	http://bind.ca/cgi-bin/dataget?get=search&rectype=4&type=int&id=%s	BIND Interaction Link
DIP Component	http://dip.doe-mbi.ucla.edu/dip/DIPview.cgi?PK=%s	DIP Node Link
DIP Reaction Link	http://dip.doe-mbi.ucla.edu/dip/DIPview.cgi?IK=%s	DIP Interaction Link
Expasy Enzyme Link	hhttp://www.expasy.org/cgi-bin/getenzyme-entry?2.7.1.1	Expasy Enzyme Link
Expasy Prosite Link	http://www.expasy.org/cgi-bin/getprosite-entry?PS00378	Expasy Prosite Link
Genpept Link	http://ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=protein&dopt= GenPept&list_uids=83035	Genpept Protein Link
IUBMB	http://www.chem.qmul.ac.uk/iubmb/enzyme/EC%d.html	Enzyme Commission
KEGG Compound	http://www.genome.ad.jp/dbget-bin/www_bget?compound+C00022	KEGG Component Link
KEGG Enzyme	http://www.genome.ad.jp/dbget-bin/www_bget?enzyme+%s	KEGG Enzyme Link
KEGG Reaction Link	http://www.genome.ad.jp/dbget-bin/www_bget?rn+%s	KEGG Reaction Link
OMIM link	http://www.ncbi.nlm.nih.gov/htbin-post/Omim/dispmim?138079	OMIM Disease Link
PDB	http://www.rcsb.org/pdb/cgi/explore.cgi?pdbId=1BDG	PDB Structure Link
PIR Sequence	http://pir.georgetown.edu/cgi-bin/nbrfget?xref=1&id=JT0482	PIR Protein Link
PROMISE	http://metallo.scripps.edu/PROMISE/%s.html	Protein Active Sites

Table 4.7 Pre-defined URLs

Description	URL	Display Name
PubMed	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=%s&dopt=Abstract	PubMed Literature
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?key=2.7.1.1	Struct. Class. Of Prot. (SCOP) Link
SwissProt Protein Entry	http://www.expasy.org/cgi-bin/niceprot.pl?P17709	SwissProt Protein Link
TransPath	http://www.biobase.de/cgi-bin/biobase/transpath/3.4_demo/bin/get.cgi?%s	TransPath
VNTI (DNA/RNA) VNTI (Protein) VNTI (Citation) VNTI (BLAST)	vnti:DNA/RNA/%s VNTI:Protein/%s vnti:CITATION/%s vnti:BLAST/%s	VNTI (DNA/RNA) VNTI (Protein) VNTI (Citation) VNTI (BLAST)

Table 4.7 Pre-defined URLs (Continued)

CHAPTER

5

DRAWING PATHWAYS

This chapter describes how to draw pathways in the PathBlazer Viewer. Many of the tasks in this chapter are described using glycolysis as an example to illustrate various functions in the context of a well known metabolic pathway.

Topics in this chapter include:

- *Introduction to Drawing Pathways* on page 109
- *Drawing Tools* on page 110
- *Drawing a New Pathway* on page 112

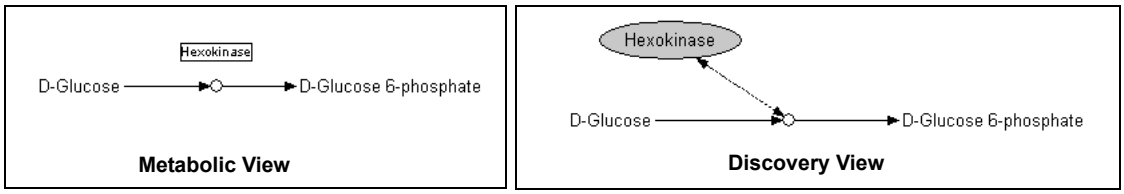
Introduction to Drawing Pathways

A key feature of Vector PathBlazer is the ability to draw known and novel pathways by combining public and proprietary data. Pathways can be drawn in the Graphics window in the following ways:

- by creating new components and connecting them into reactions
- by adding existing components in the database and connecting them into reactions
- by adding existing reactions in the database
- by adding existing pathways in the database

Pathways can be drawn in two different kinds of modes or views: **Metabolic** and **Discovery**. The main difference between these two views is how catalyzing agents (that is, enzymes) and protein-protein interactions are displayed.

- In **Metabolic View**, the enzyme is not graphed as a separate element and the reaction that includes the enzyme is not graphed as a separate connector. Instead, the enzyme is drawn as a label of the reaction node. The enzyme is still an independent object in the database and is selected from the database but is displayed close to the reaction node.
- In **Discovery View,** the enzyme is drawn as a separate component of the reaction and is connected to the reaction node by a double-headed arrow to indicate that the enzyme is catalyzing the reaction.



Note: Protein-protein interactions can only be drawn in **Discovery View**.

Drawing Tools

The Palette window contains a set of drawing tools that include shapes for representing components and lines for representing connectors (*Figure 5.1*).

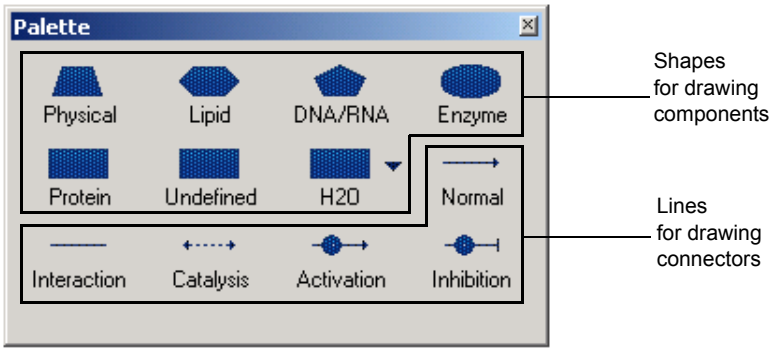


Figure 5.1 Palette of drawing tools

Component Shapes and Connector Lines

Shapes and lines in the Palette window can be used to represent any kind of molecule or interaction (for example, protein, DNA, etc.) and are labeled to suggest a template for their use. For example, the oval is labeled *Enzyme* to suggest that each time you draw an enzyme, you use an oval. Components and connectors are automatically assigned the type suggested by their labels. However, once a shape is created, assigned a name, and saved to the database, you can change the shape in the Graphics window without changing the type associated with the shape. You can permanently change the type by modifying it in the object's annotations. For more information about annotating objects, see *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37. Available shapes and their suggested uses are shown in the following table (Table *5.1*).

Shape	*Suggested Use*
trapezoid	Physical factor Example: heat, light, etc
hexagon	Lipid
pentagon	DNA/RNA
ellipse	Enzyme

Table 5.1 Shapes in the Palette window

Shape	Suggested Use
rectangle	Protein Unidentified molecule

Table 5.1 Shapes in the Palette window (Continued)

Available connectors and their suggested uses are shown in the following table (Table 5.2)

Connector	Suggested Use
	Unidirectional reaction that can be used to indicate a left to right or a right to left reaction direction. **Note:** To create a reversible reaction, two separate and opposite reactions are created using this connector.
	Protein-protein interaction **Note**: A straight line automatically confers protein-protein interaction on a reaction and only displays when drawing in Discovery/Unrestricted View.
	Catalysis reaction **Note**: This line only displays when drawing in Discovery/Unrestricted View.
	Activating reaction
	Inhibiting reaction

Table 5.2 Connectors in the Palette window

Commonly Used Molecules

In addition to the shapes that can be used to represent any kind of molecule, a list of commonly used molecules is provided by the drop-down menu next to the

symbol in the Palette window. A number of small molecules such as H_2O and ATP have already been created as components in the default database that is installed when Vector PathBlazer is installed and can be further annotated to suit your needs. Each small molecule references the corresponding component in the database by primary name. The drop-down list includes the following small molecules:

- H_2O
- NAD^+
- NADH
- $NADP^+$
- NADPH
- Oxygen
- Orthophosphate
- CO_2
- H^+
- FAD

- ATP
- ADP
- $FADH_2$

Note: When a new database is created, the list of small molecules above is automatically created in the new database.

***Add a component to the list*—**by selecting **Tools > Options** and clicking the **Set Palette Pull-Down Molecules** tab in the **Options** dialog box that opens (*Figure 5.2*). Click **Add**, enter its name in the dialog box, and click **OK**. The component is added to the list.

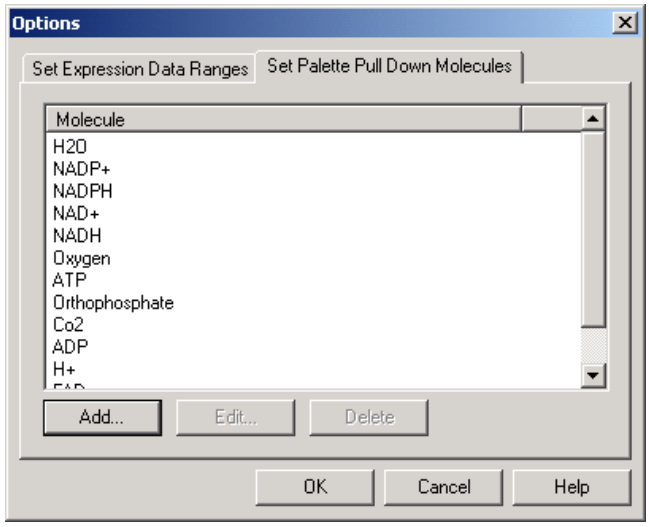


Figure 5.2 Tab in the Options dialog box where components are added to the common molecules list

***Edit the name of a component in the list*—**by selecting the component, clicking **Edit**, and changing the name.

Note: Only primary names of components in the database and not synonyms can be added to the **Common Molecules** list. If the component added to the list is not already present in the database, you will be able to add it to the list but you will receive an error when you try to draw the component in the Graphics window. Add it to the database by importing it or by drawing it in the Graphics window and saving it to the database. The same is true if you edit the name of a component to one that is not present in the database.

***Delete a component from the list*—**by clicking **Delete**. The component is removed from the list only; it is not removed from the database.

Drawing a New Pathway

There are several ways to draw new pathways in Vector PathBlazer. Use the steps outlined in the following sections to draw a new pathway.

Opening A New Graphics Window

Use the following steps to open a new Graphics window.

1. Select **File > New >** and select one of the submenus: **Metabolic Pathway** or **Discovery Pathway**. You can also click the **New Pathway** button ( ) on the toolbar and select from one of the submenus off of the drop-down menu next to the button.
2. A blank Graphics window opens that is labeled at the top of the window with either **New Pathway1 [Metabolic (restricted) view, Database]** or **[Discovery (unrestricted) view, Database]**. **Database** indicates that the pathway is stored in the Vector PathBlazer data-

base as opposed to a .pw file. For information about saving pathways to .pw files, see *Saving a Pathway or Reaction to the Database or a File* on page 46. Continue to the next sections to add components and reactions to the Graphics window.

Adding a Component

You can add any number of components and connectors in the Graphics window to form any number of reactions in a pathway. A set of reactions in the Graphics window represents one pathway. You can add components to pathways by:

- drawing a new component
- drawing an existing component
- selecting a component from the Database Explorer

Drawing a New Component

When you draw a new component and name it, Vector PathBlazer first searches the database for any components with the same primary name or with the same synonym. For example, a common synonym of hexokinase is glucokinase. If you wanted to create this enzyme in the database by drawing it and you entered the name ***glucokinase***, Vector PathBlazer would search the database for *glucokinase*. Three options can be returned from the search:

1. The program finds a component in the database that has the primary name glucokinase and names the shape (that is, the component) drawn in the Graphics window ***Glucokinase***.

Reminder: You have not created a new component by drawing it and then selecting its name from the database; you are simply referencing a component that was already in the database.

2. The program finds a synonym component in the database that has the primary name hexokinase and the synonym glucokinase. You are offered the option to make ***glucokinase*** the default name in the database, or alternatively to make ***glucokinase*** the display name within the current pathway only.

   If you decide to leave the default name for the component, ***hexokinase***, you can change the name later by selecting it, then choosing **Change Component Display Name** from the shortcut menu. In the dialog box that opens, you must select from among the displayed identities (synonyms) currently in the PathBlazer database. You cannot assign any other name to the component. In the dialog box, you can specify whether the new display name is for the current pathway only, or to be displayed in all pathways henceforth.

3. The program does not find a component named glucokinase by primary name or synonym, names the shape Glucokinase, and creates the corresponding component in the database.

If the program determines that the component is not already in the database, it opens a wizard that assists you in creating the new component including naming and annotating it. Use the following steps to draw and annotate a new component.

1. Select a shape in the Palette window and move the cursor to the Graphics window. The cursor changes to the symbol . Click anywhere in the Graphics window to insert the shape. When the shape is initially inserted, it is called <UNNAMED> by default.

Note: The cursor remains a wand until you either click another shape or line in the Palette window, click on one of the buttons in the Graphics toolbar such as the arrow icon ( ), or press ESCAPE.

2. To assign a name to the shape, click the arrow icon on the toolbar, double-click on the shape, enter a new name, and press enter. If the name matches an object in the database by primary name or by synonym, the object is automatically named by the primary name. If the entered name does not match an object already in the database or a synonym of an object, (see preceding page) a dialog box opens allowing you to select among several

options related to naming the new shape (*Figure 5.3*). (Only options appropriate for your new object-type are available.)

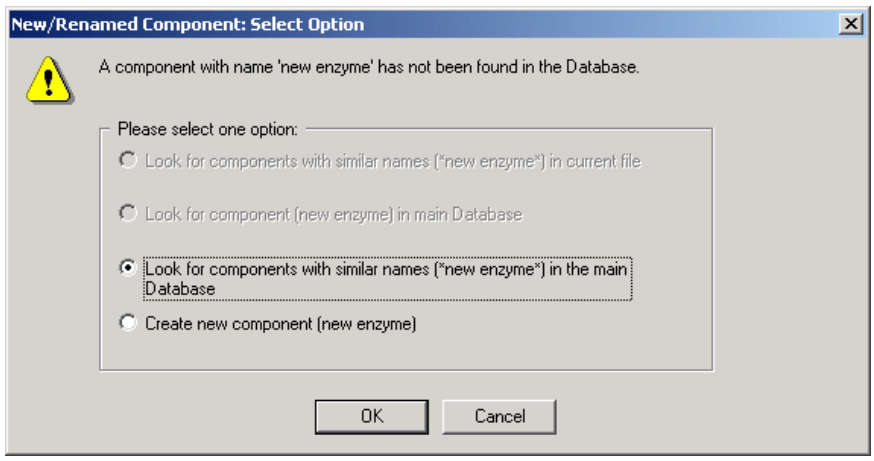

New/Renamed Component: Select Option
A component with name 'new enzyme' has not been found in the Database.
Please select one option:
Look for components with similar names (*new enzyme*) in current file
Look for component (new enzyme) in main Database
Look for components with similar names (*new enzyme*) in the main Database
Create new component (new enzyme)
OK
Cancel

Figure 5.3 Prompt to search the database for a component

The first three radio buttons allow you to search the database for an existing object(s). More information about those options are provided in the next section. To create a new object, select the **Create a new component...** radio button. Click **OK**. This opens the Component wizard.

3. The Component wizard contains a list of screens that allow you to name and add annotations to a component when you are creating it.

   - If you do not want to add annotations, simply name the component in the third screen. Continue to step through the wizard using the **Next** button in each screen until the **Finish** button displays. Click **Finish** to create the component. Once the component is saved to the database, you can annotate it at any time.

   For a description of each annotation field and its values, see *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37.

   - To add annotations, in the first screen of the **Component** wizard, select the **Create new component** radio button and click **Next** (*Figure 5.4*).

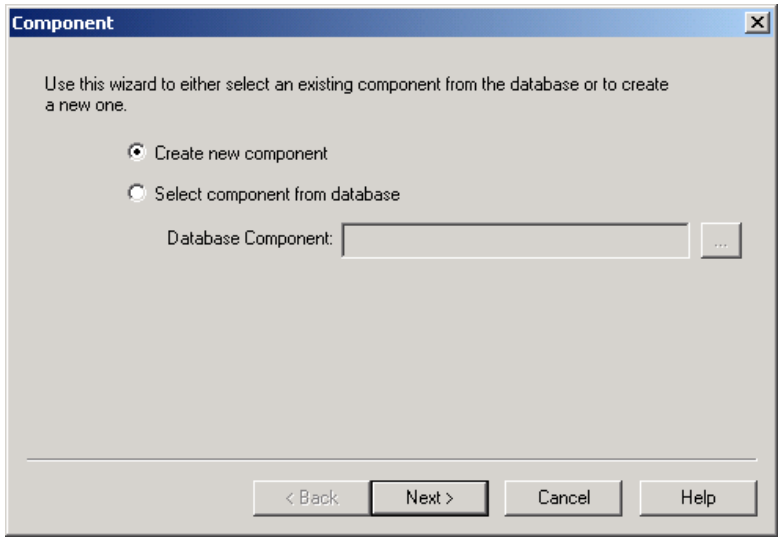


Figure 5.4 Wizard for creating new components

4. Enter general information to name and describe the component in each of the fields (*Figure 5.5*). Only the **Name** field is a required field. Click **Next**.

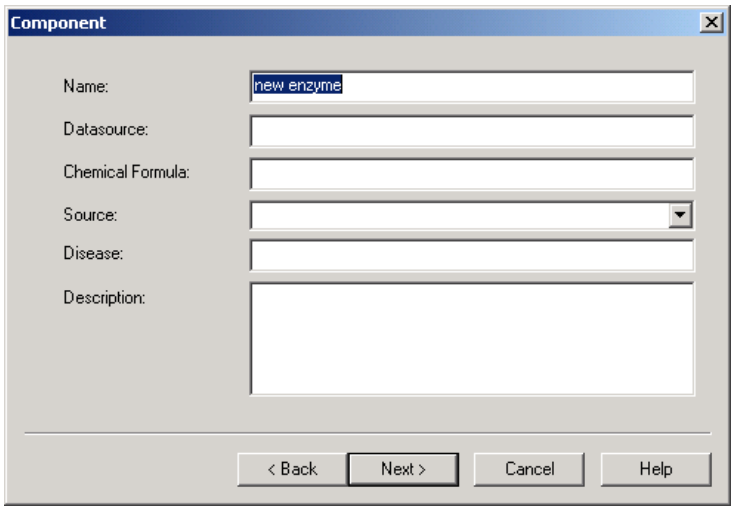


Figure 5.5 Wizard for creating new components: adding general information

5. Select the Component Class from the drop-down menu (*Figure 5.6*). Available fields differ depending on which type of component is selected from the **Component class** field. Enter information about the component's type in each of the fields. The type, and subtype, if available, are automatically entered depending on the selected shape. Click **Next**.

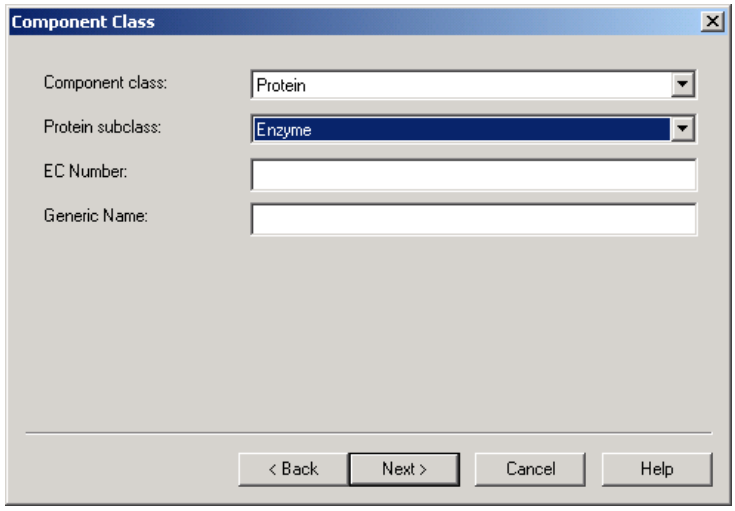


Figure 5.6 Wizard for creating new components: defining a component's type

6. Enter information about the component's location (*Figure 5.7*) or organism source. Click the **Add** button to add a location or organism source and fill in the **Type**, **Tissue**, and **Subcellular Location** and **Name** fields (*Figure 5.7*). More than one location can be added by clicking **Add** to add each additional location. Once a row is added, click **Edit** to change the informa-

tion or **Delete** to delete the row. Click **OK**. Click **Next**, and add **Organism** in the same manner..

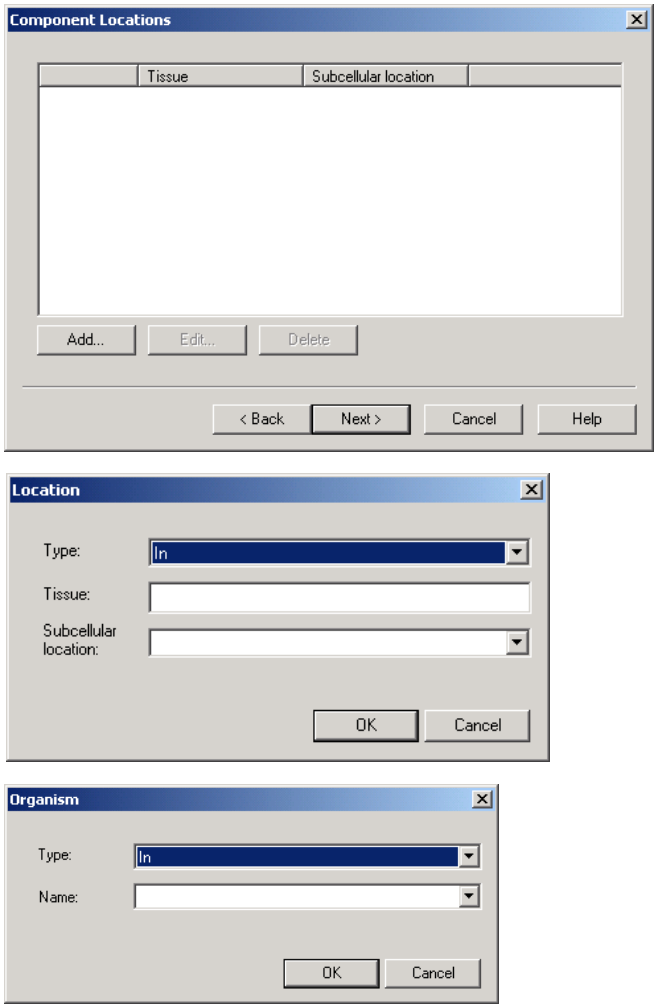


Figure 5.7 Wizard for creating new components: describing a component's location, source tissue and organism

7. Enter any crosslinks to a component (*Figure 5.8*). A crosslink is a link to either the Vector NTI database or to an external database. Click **Add** and enter information in the **Type**

(either database or URL), **Database** (for example, VNTI (DNA/RNA), and **Accession ID** fields. Click **OK**. Click **Next**.

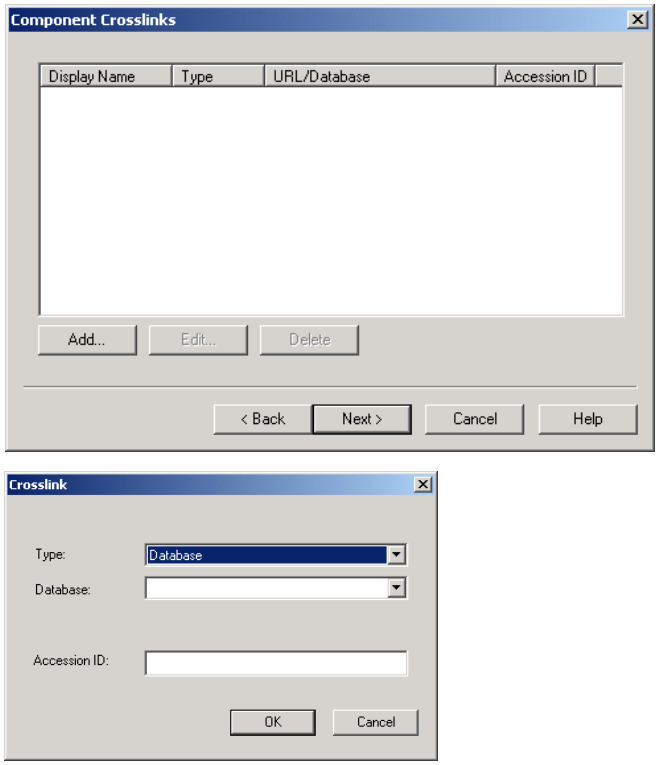


Figure 5.8 Wizard for creating new components: adding database crosslinks

8. Enter any synonyms that are associated with the component one at a time (*Figure 5.9*). Click **Add**, enter a synonym name in the dialog box, and click **OK**. Click **Add** to add another synonym. Click **Finish**.

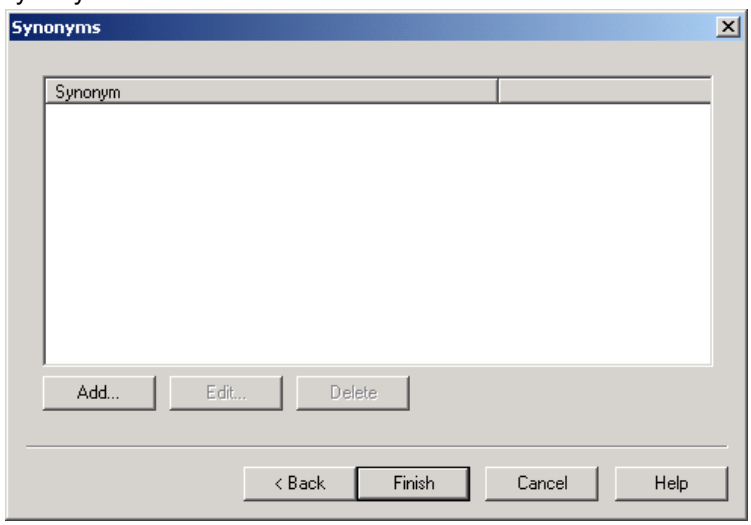


Figure 5.9 Wizard for creating new components: adding synonyms

9. The component is named in the Graphics window and is saved to the database with any annotations. To change the graphical properties of an object (for example, font color and size), see *View and modify an object's graphical properties* on page 19.
10. If you choose to select a component from a database, press the **Browse** button (...) to locate the component in the existing database.

Drawing an Existing Component

You may have components already in the database that you either created by import or by drawing *de novo.* You can access a component from the Graphics window by first drawing a shape to represent it and then searching the database to name the component and provide any annotations that have already been attributed to it. Adding a component this way is useful if, for example, you have drawn a component and assigned a set of graphical properties to it and then want to overlay the components annotations on the shape.

Adding an existing component with the drawing tools is similar to adding a new component except, to name the component, you search the database for the component you want to add and then annotate it further. Use the following steps to draw a component and then search the database to name it.

1. Select a shape in the Palette window and move the cursor to the Graphics window. The cursor changes to the symbol *□. Click anywhere in the Graphics window to insert the shape. When the shape is first inserted, it is called <UNNAMED> by default.

Note: The cursor remains a wand until you either click another shape or line in the Palette window, click on one of the buttons in the Graphics toolbar such as the arrow icon ( ), or press ESCAPE.

2. To assign a name to the shape, click the arrow icon on the toolbar, double-click on the shape, enter a name, and press ENTER. If the name matches an object in the database by primary name or by synonym, the object is automatically named by the primary name. If the entered name does not match an object already in the database or a synonym of an object, a dialog box opens, allowing you to select among several options related to naming the new shape (*Figure 5.10*). (Only options appropriate for your new object-type are available.)

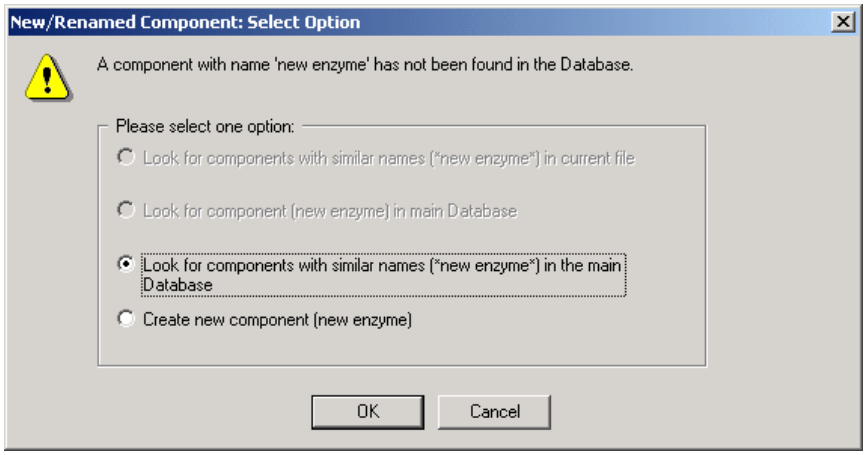

New/Renamed Component: Select Option
A component with name 'new enzyme' has not been found in the Database.
Please select one option:
Look for components with similar names (*new enzyme*) in current file
Look for component (new enzyme) in main Database
Look for components with similar names (*new enzyme*) in the main Database
Create new component (new enzyme)
OK
Cancel

Figure 5.10 Select the preferred option for naming or renaming a new component

3. The first three radio buttons allow you to search the database for an existing object(s). More information about those options are provided in the next section. To draw an existing componet, choose **Look for Components with Similar Names**, then click **OK**.
4. If the program can match the entered name to any components in the database by primary name or by synonym, the **Select a component** dialog box displays any potential matches. The search is performed as a string search by primary name and synonym and all partial matches display. For example, if '**glu**' is entered with the intention of finding 'glucose' then the list in *Figure 5.11* is returned. Note that components such as '**Glucagon**' and '**Glucose 1-phosphate**' are returned in addition to 'Glucose'.

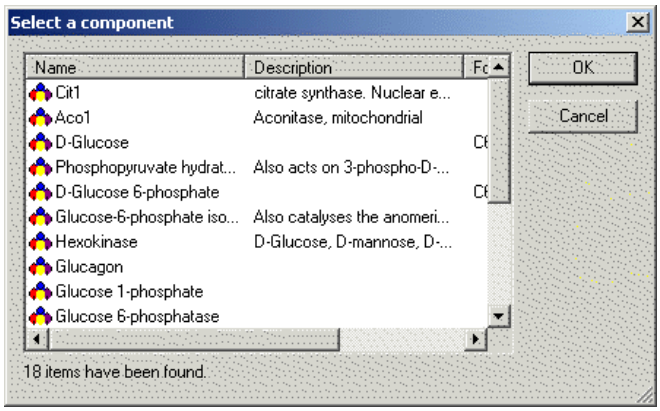


Figure 5.11 Multiple matches can be returned when a name is entered as a partial string

If the program cannot match the entered name to any components in the database, a PathBlazer message informs you that the component was not found.

You can open the Component wizard directly by selecting the newly drawn component and then selecting **Component Properties** from the shortcut menu. However, when the database is searched, objects are only searched by primary name and not also by synonym.

In the first screen of the Component wizard (*Figure 5.12*), select the **Select component from database** radio button and click the **Browse** button (...).

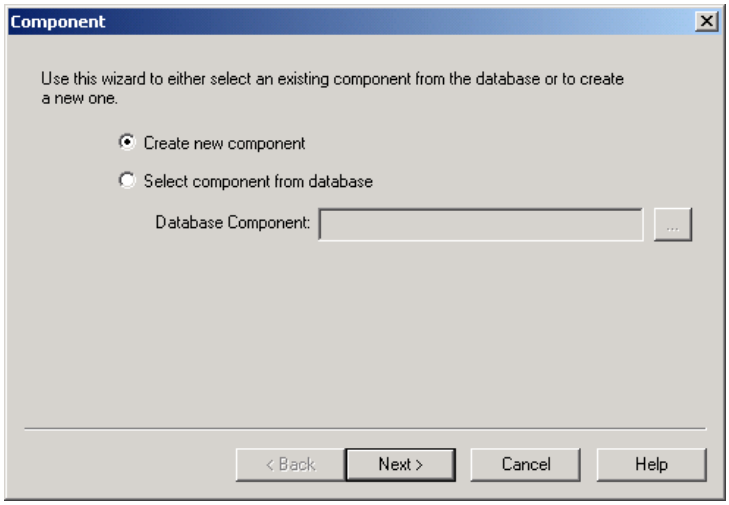


Figure 5.12 Wizard for selecting and creating new components

5. A list of subsets displays in the **Open** dialog box. Select the subset to search the component you are looking for and double-click on it or click **Open** (*Figure 5.13*). To search all components in the database, select the **All Components** subset.

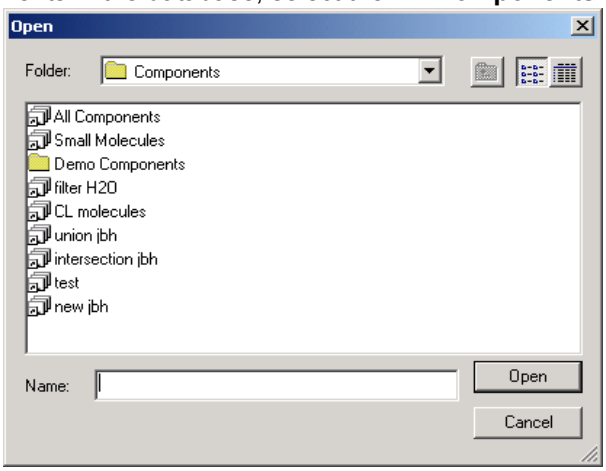


Figure 5.13 Dialog box for selecting subset to search for component

6. In the next dialog box, select the component you are looking for from the components that display. To search a different subset, click the button and select a different subset. Click **Open**. The selected component is entered in the **Database Component** field in the Component wizard (*Figure 5.14*). Click **Next**.

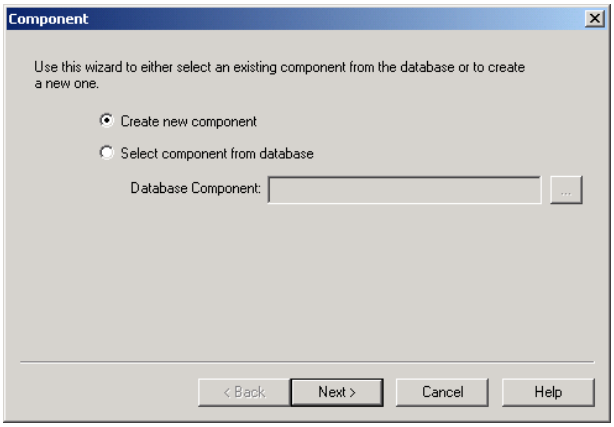

Component
Use this wizard to either select an existing component from the database or to create a new one.
Create new component
Select component from database
Database Component:
< Back
Next >
Cancel
Help

Figure 5.14 Component wizard with Database Component selected

7. The remaining screens in the Component wizard are for adding annotations to a component, which may or may not already be annotated. If you do not want to add annotations, click **Next** in each screen until the **Finish** button displays in the last screen. Click **Finish** to name the component in the Graphics window. The Annotation screens are the same as those described in *step 4.* on page 115 through *step 8.* on page 117. For a description of each annotation field and its values, see *Annotating Pathways, Components, Experiments, Reactions, and Connectors* on page 37.

8. When you have finished adding annotations, the component is named in the Graphics window based on the selected component (*Figure 5.15*). Any annotations that were changed are also saved to the database. To change the graphical properties of a component (for

example, font color and size), see *View and modify an object's graphical properties* on page 19.

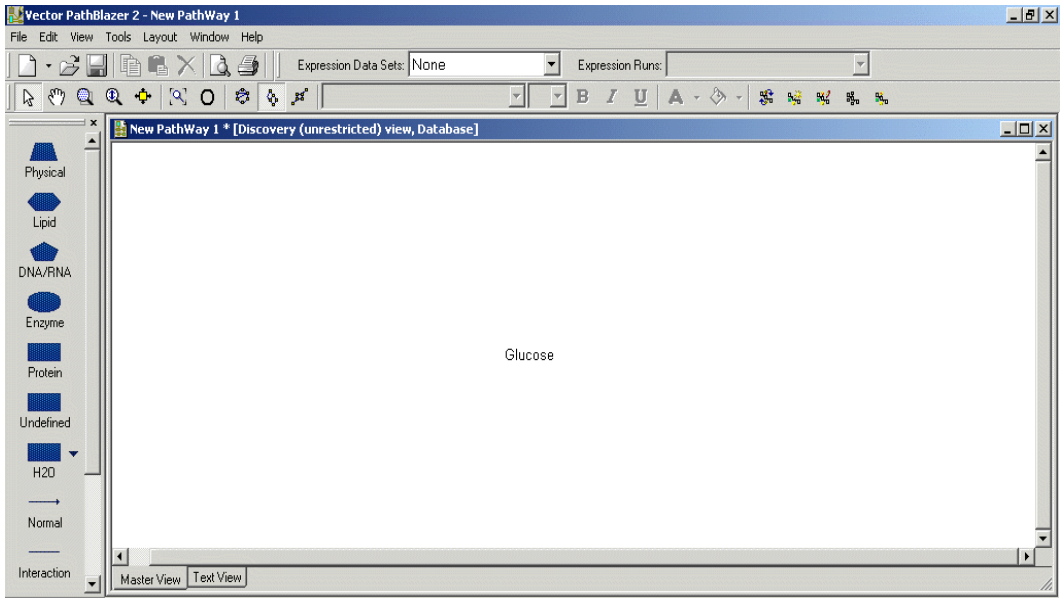


Figure 5.15 Drawing a new component

Adding An Existing Component from the Database Explorer

You can drag and drop any component that is already present in the database directly from the Database Explorer onto the Graphics window to add the component to a reaction or pathway. For example, you might have drawn the components that are involved in the glycolysis pathway *de novo* and saved them to the database and then you want to reuse some of these components to draw the gluconeogeneis pathway.

1. To add a component this way, locate the component you want to add to the reaction or pathway in a subset of the **Components** folder in the Database Explorer.
2. Select it and drag it into the Graphics window. The component is added to the Graphics window in the location where you dropped it and its name displays (*Figure 5.16*). If a component has a type associated with it, such as Enzyme, then the appropriate graphical properties associated with it display, such as an oval.

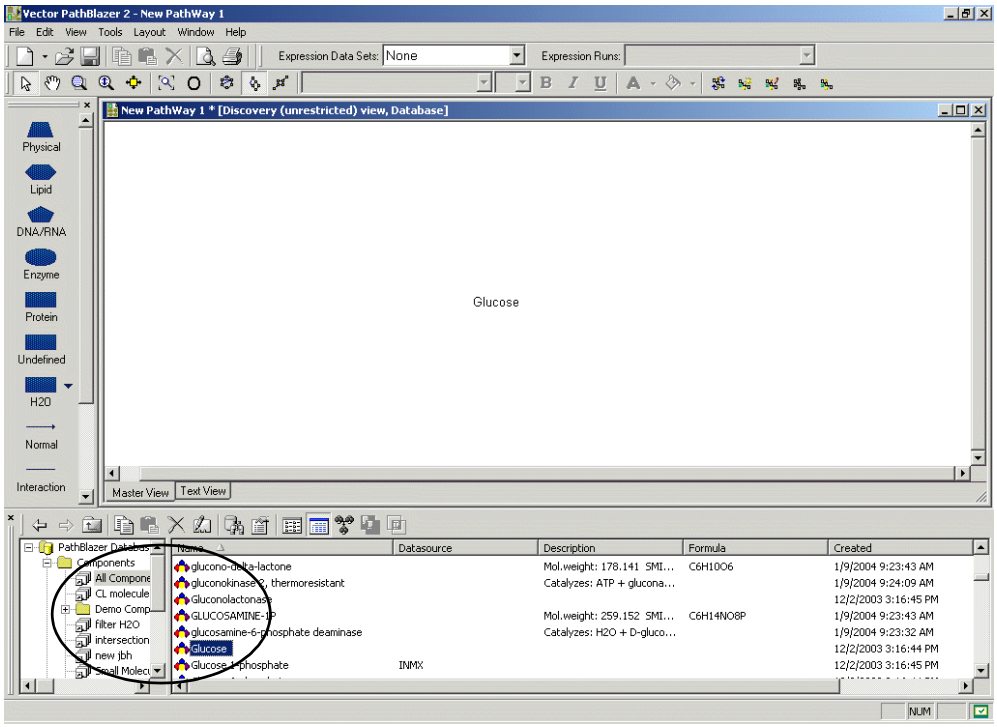


Figure 5.16 Adding a component from the Database Explorer

3. Add as many components to a single Graphics window as required. Continue to the next section to link components into reactions.

Adding a Reaction

Similar to components, reactions can either be drawn *de novo* or existing reactions can be added from the database. You can add an unlimited number of reactions to a single pathway and each reaction does not necessarily have to be joined together. For example, you might want to represent all of the protein-protein interactions in a pathway from the BIND database, where each interaction is not necessarily linked to a subsequent interaction. Instead the pathway is made up of a number of separate protein-protein interactions.

Drawing a New Reaction

Components are joined into reactions by connectors, which are represented as lines in the Palette window. At least two components must be present in the Graphics window before a connector can be added. Use the following steps to join components into reactions.

1. If there is only one component in the Graphics window, add at least one more using one of the methods described in *Adding a Component* on page 113.
2. To add a connector between two components to create a reaction, select a line from the Palette window and move the cursor to the Graphics window where it changes to a wand ( ). Click on the first component you want to link, drag the wand to the second component, and click on the second component.

The connector is drawn between the two components (*Figure 5.17*). Once two components are linked a reaction is formed between the two and is represented by a reaction node ( ).

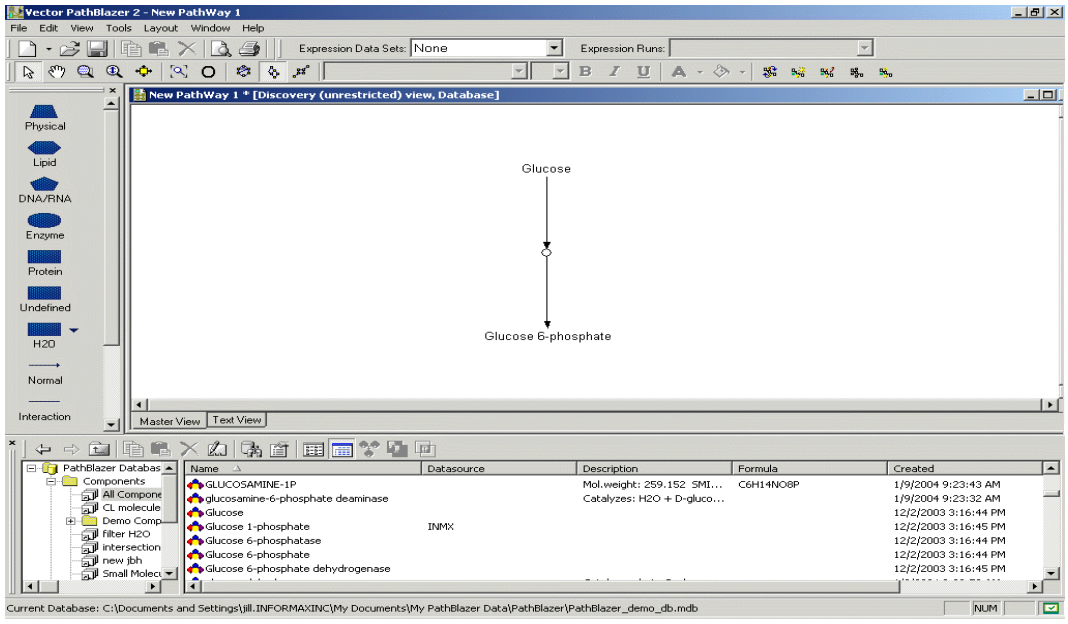


Figure 5.17 Connecting two components into a reaction

Note: The cursor remains a wand until you either click another shape or line in the Palette window, click on one of the buttons in the Graphics toolbar such as the arrow icon ( ), or press ESCAPE.

3. When multiple components are involved in one reaction, additional components are linked directly to the reaction node. You can think of the reaction node as a “hub” where one to many components can lead into it and one to many components can result from it. For example, when hexokinase mediates the transfer of a single phosphate from ATP to glucose to form glucose-6-phosphate and ADP, all of these components lead to or result from the same reaction node. Therefore, once the first two components are drawn to create a reaction node, the remainder of the components can be drawn to the reaction node itself (*Figure 5.18*).

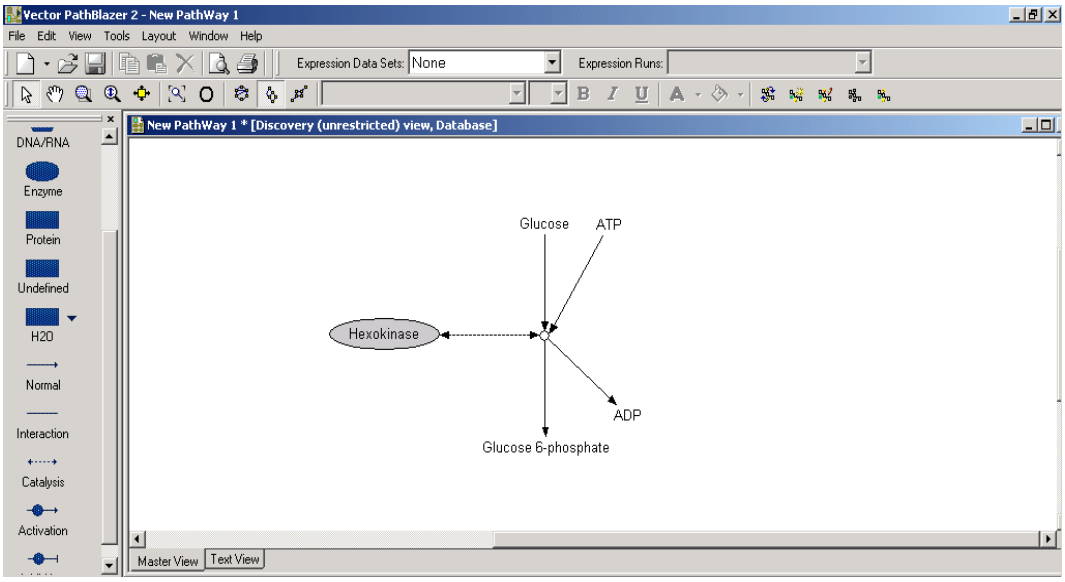


Figure 5.18 Many components are joined into a reaction via a single reaction node

4. Continue adding components and connectors to a single reaction node or add additional components and connectors to form other reactions. You can add multiple reactions to a pathway without joining each reaction in the pathway. To connect two reactions into a pathway, join the ending or resulting component of one reaction with the starting component of the next. In *Figure 5.19*, the first and second steps of glycolysis are joined to form a pathway via the component Glucose-6-Phosphate, which then becomes part of two different reactions.

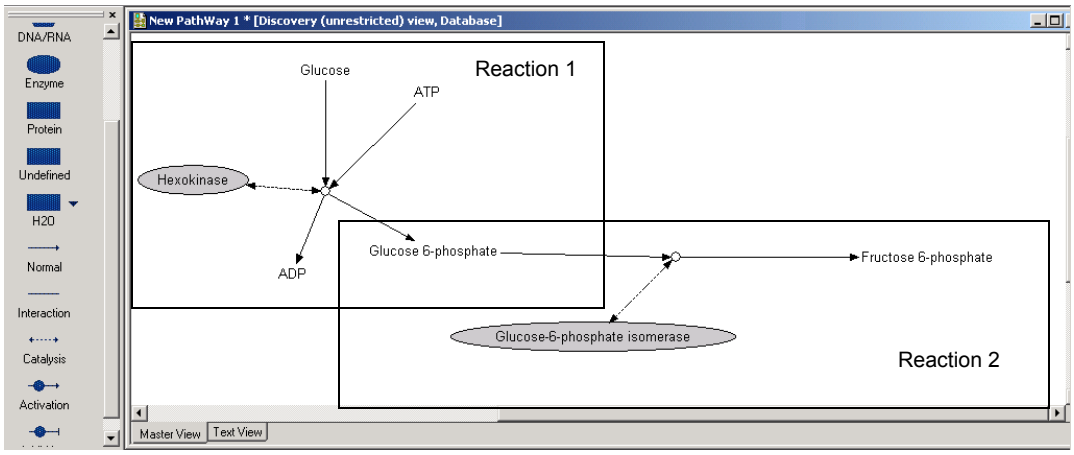


Figure 5.19 Joining two reactions into a pathway

5. When you form reactions using connectors, the reactions are called <UNNAMED> by default and are not saved to the database automatically. The pathway is also not saved automatically. For instructions on how to save pathways and reactions, see *Saving PathBlazer Components, Reactions and Pathways* on page 46.

Note: You can change the type of connector (for example, change an inhibition to an activation) by clicking on the line representing the connector and selecting **Object Properties** from the shortcut menu. In the Object Properties box, select a different line style from the drop-down list in the **Style** field and close the box. The new style is applied to the connector.

Adding an Existing Reaction from the Database Explorer

Any reaction stored in the database can be added to the Graphics window directly from the Database Explorer. In the Database Explorer, locate a reaction you want to draw in the Graphics window and select **Open** from the shortcut menu or double-click on the reaction. The reaction and all components in the reaction display in the Graphics window (*Figure 5.20*).

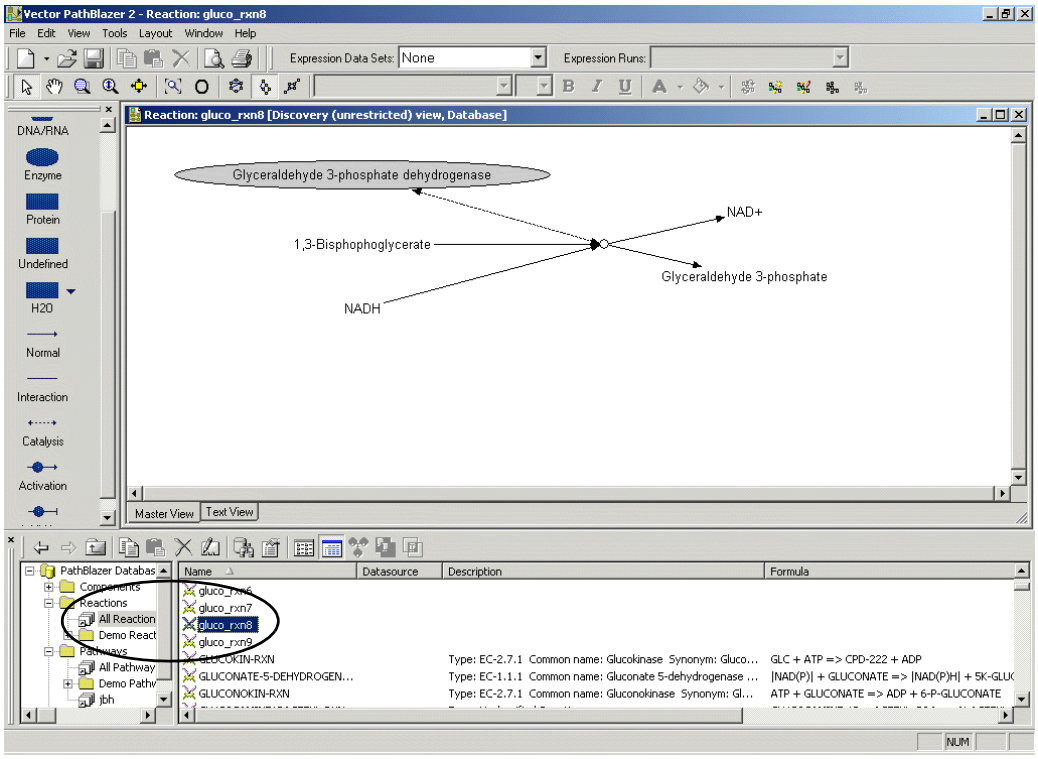


Figure 5.20 Adding a reaction from the Database Explorer

You can add only one reaction to a Graphics window using this method. If you double-click on a second reaction, a new Graphics window opens. You can add additional components by drawing them from the Palette window or dragging them from the Database Explorer and then joining them to components in the reaction you opened. You can also add reactions using the method described in the following section.

Adding an Existing Reaction from the Graphics Window

Reactions can be added from the database that have a component in common with one you have selected in the Graphics window. For example, you might have opened the first reaction in glycolysis using the method described in the previous section and now you want to add the second reaction that starts with glucose 6-phosphate without having to draw components and connect them.

To add a stepwise reaction, all reactions in the database or in one or more specified subset(s) that have a component in common with the selected component are searched and presented in a list. To add a reaction using this method, use the following steps.

1. Select a component in the Graphics window and then select **Add reaction** from the shortcut menu.

Note: When adding a reaction by this method, components are only searched by primary name and not by synonym. If the component you have selected matches any other components by synonym, those reactions are not displayed in the returned list.

2. In the **Add Reactions** dialog box, select the direction in which you want the selected component to participate in any matching reactions (*Figure 5.21*). Select from the options in the drop-down list in the **Role of** field: **Input/PPI**, **Output**, or **Catalyzing agent**. Consider the options in the context of the following reaction: glucose + ATP + hexokinase > glucose-6-phosphate + ADP.
    - **Input/PPI** means any reaction that includes the selected component as either an input to a reaction or part of a protein-protein interaction (since these types of interactions are non-directional). If glucose were the selected component, the reaction above would be returned since glucose is an input to the reaction.
    - **Output** means any reaction that includes the selected component as an output of a reaction. If glucose-6-phosphate were the selected component, the reaction above would be returned since glucose-6-phosphate is an output of the reaction.
    - **Catalyzing agent** means any reaction in which the selected reaction participates as the catalyzing agent. If hexokinase were the selected component, the reaction above would be returned since hexokinase is the catalyzing agent of the reaction.
3. In the **Select Subset Search** field, navigate to one or more reaction subsets. Select the checkbox next to each subset and click **Search**.

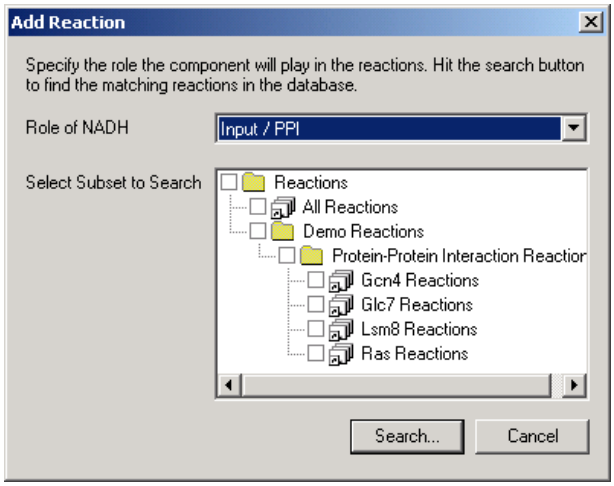


Figure 5.21 Specifying the direction in which reactions should be searched for a selected component

4. In the next dialog box, all matching reactions that contain the selected component participating in the specified direction are listed (*Figure 5.22*). Information about the components displays in three columns: Reaction, Generality, and Formula. The **Reaction** column displays the name of the reaction. The **Generality** column lists the Interaction Generality (IG) value for reactions that are protein-protein interactions. For reactions that are not protein-protein interactions, a hyphen displays in this column. The **Formula** column lists the participating components in a reaction and the reaction direction.

To see more details about a reaction, slide the divider bars of any column to the left or right to make a column larger or select a reaction and select **Properties** from the shortcut menu.

Select one or more reactions to add to the Graphics window by selecting the checkbox next to each reaction in the Reaction column and click **OK** (*Figure 5.22*).

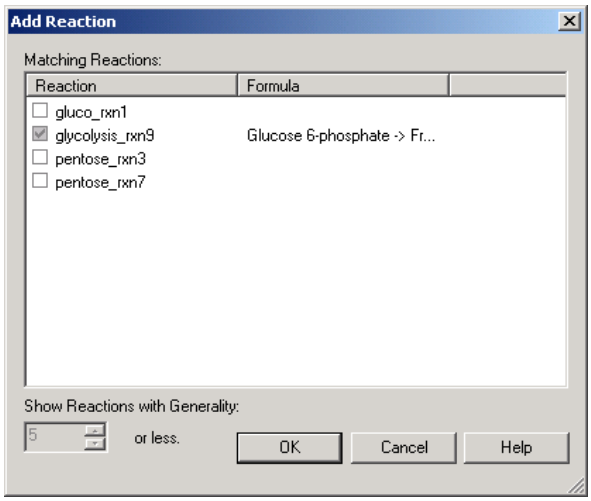


Figure 5.22 List of reactions returned that match a selected component by primary name and direction

5. The reaction is added to the pathway by joining it to the selected component (*Figure 5.23*).

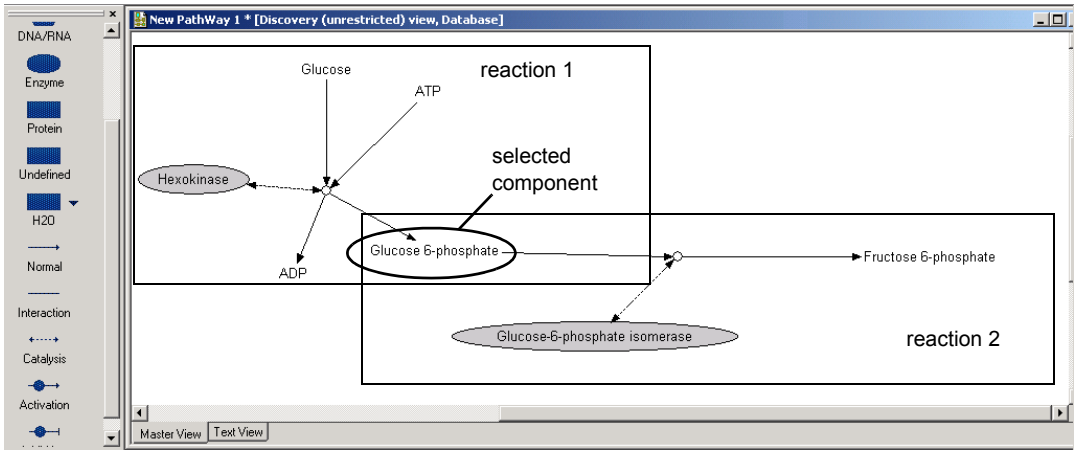


Figure 5.23 Reactions joined by a selected component

Changing a Saved Reaction

To store a reaction in a pathway and save it as an independent object, the reaction must be saved to the database. When a reaction is saved, associations to the connectors and components to which it is linked are saved with it. For information about saving reactions, see *Saving PathBlazer Components, Reactions and Pathways* on page 46. When a component or connector is changed or deleted or when a component is added to a saved reaction, you are prompted

to either update, create, or disconnect the reaction from the pathway. Table *5.3* describes each action.

Action	*Description*
Update reaction	Makes the change to the reaction in the pathway and resaves the reaction under its original name when the pathway is saved. This option is only available when the reaction does not participate in more than one pathway in the database. This option is also only available when an added component already exists in the database.
Create new reaction	Makes the change to the reaction, appends the name of the reaction with an incremental number, and saves it as a new reaction in the database when the pathway is saved. The new reaction takes the place of the original reaction in the pathway. Any annotations that were present in the original reaction are retained. This option is only available when an added component already exists in the database.
Disconnect this reaction from pathway	Makes the change to the reaction but disconnects the current reaction from the pathway and adds an unsaved reaction to the pathway called <UNNAMED>. Also, any annotations in the original reaction are not applied to the new, unnamed reaction. This is the only option available when an unnamed component is added to a reaction.

Table 5.3 Options when a component is added or a connector is changed in a reaction

Adding a Component to a Saved Reaction

To add a new component to a reaction that has already been saved to the database, use the following steps.

1. Add a component by selecting a shape from the Palette window or dragging a component from the Database Explorer. Connect it to a reaction node with a line from the Palette window to form a connector between the new component and the reaction node. When a connector links the newly added component to the reaction node the dialog box described in the next step automatically opens. The actions available in the dialog box depend on whether a component is named or unnamed in the Graphics window.
2. If the added component is unnamed (that is, a name has not yet been assigned to a newly drawn shape in the Graphics window), the dialog box in *Figure 5.24* displays with only one available option in the Action box: **Disconnect the reaction from the pathway**.

'Disconnect' refers to the current reaction because in order for a component to be added to a saved reaction, the component must be named. Therefore, an unnamed reaction takes the place of the original reaction.

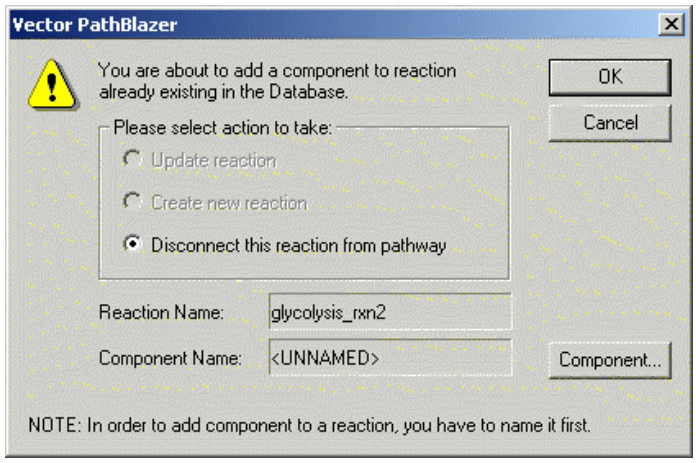

Vector PathBlazer
You are about to add a component to reaction already existing in the Database.
OK
Cancel
Please select action to take:
Update reaction
Create new reaction
Disconnect this reaction from pathway
Reaction Name:
glycolysis_rxn2
Component Name:
<UNNAMED>
Component...
NOTE: In order to add component to a reaction, you have to name it first.

Figure 5.24 Option when adding an unnamed component to a saved pathway

The original reaction name (for example, glycolysis_rxn2) displays in the **Reaction Name** field and the Component name (<UNNAMED>) displays in the **Component Name** field. To disconnect the reaction from the pathway, click **OK**. Since a reaction saved to the database cannot contain any unnamed components, the original reaction is disconnected from the pathway and a new reaction called <UNNAMED> that contains the newly added connector and <UNNAMED> component is added in its place. Additionally, <UNNAMED> reactions cannot be saved to the database. The reaction that now displays in the Graphics window is the <UNNAMED> reaction. The original reaction (glycolysis_rxn2 in this example) remains unchanged in the database but is no longer connected to the pathway.

To name the component from the dialog box, click **Component** and name the component by following the instructions in either *Drawing a New Component* on page 113 or *Drawing an Existing Component* on page 118. Once a component is named, the actions in the dialog box update. Continue to the next step.

3. If the added component is named in the Graphics window (that is, the component has either been named based on an existing database component or a new name has been entered in the database), the dialog in *Figure 5.25* displays with three options: **Update reaction**, **Create new reaction**, and **Disconnect this reaction from the pathway**.

Note: The option **Update reaction** is only available if the reaction does not participate in more than one pathway.

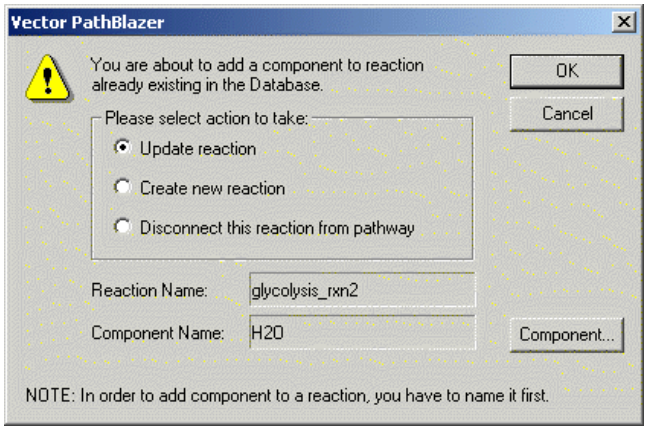


Figure 5.25 Options when adding a named component to a saved pathway

The reaction name (for example, glycolysis_rxn2) displays in the **Reaction Name** field and the Component name (H2O) displays in the **Component Name** field. Select the radio button that corresponds to the action you want to apply and click **OK**. See Table *5.3* for action descriptions.

To change the component that is being added, click **Component** and name the component by following the instructions in either *Drawing a New Component* on page 113 or *Drawing an Existing Component* on page 118.

4. To save the change, see *Saving PathBlazer Components, Reactions and Pathways* on page 46. To cancel the change and revert to the previous pathway, close the pathway without saving it and then reopen it.

Adding Selected Components or Reactions to a Subset

To add components or reactions you have selected in the Graphics Window to a subset, select the objects, then right click anywhere in the Graphics Pane.

- To save to an existing subset, select **Append Selected Components [Reaction] to a Subset.** In the Append to Subset dialog box that opens, select the subset to store the components [reactions].
- To save the objects to a new subset, click **Save Selected Components [Reactions] as a Subset**. The dialog box is similar to the Append to Subset dialog, but text boxes are available for you to name (create) and describe the new subset.

Click **Append** or **Create** to execute the command.

Deleting Components in a Reaction

When components are deleted from a saved reaction, a dialog box displays listing each reaction in which a component participates.The difference between deleting and adding components is that once a component is added to a pathway, it can participate in more than one reaction (for example, it can be a substrate in one reaction and a product in another). During deletion, actions can be applied independently to each with the same options.

To delete a component from a saved reaction, use the following steps.

1. Select the component in the Graphics window and click the **Delete** button on the Graphics toolbar or press the DELETE-key. The dialog box in *Figure 5.26* displays. If the component participates in more than one reaction, each reaction displays in a different row. Each reaction's name displays in the **Reaction** column.

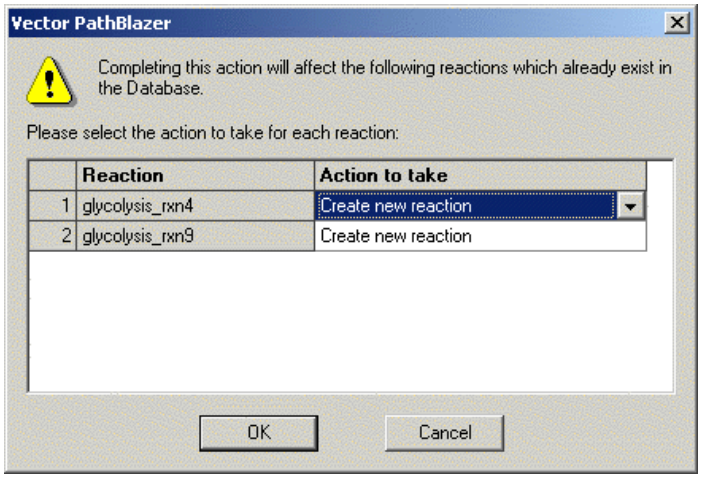


Figure 5.26 Dialog box that displays when deleting a component from a saved reaction

2. For each reaction, select an option from the drop-down list in the **Action to take** column. See Table *5.3* for action descriptions.

Note: If a reaction participates in more than one pathway, the option **Update reaction** is not available.

3. Click **OK**. The selected action is applied to each reaction listed.
4. To save the change, see *Saving PathBlazer Components, Reactions and Pathways* on page 46. To cancel the change and revert to the previous pathway, close the pathway without saving it and then reopen it.

Changing or Deleting Connectors in a Reaction

A change to a saved reaction is also triggered when a connector's annotations are changed or when a connector is deleted. To change a connector's annotations or delete a connector in a saved reaction, use the following steps.

1. Delete a connector by selecting it in the Graphics window and then clicking the **Delete** button on the Graphics toolbar or pressing the DELETE-key.

   Change one or more connector annotations by selecting a connector in the Graphics window and selecting **Connector Properties** from the shortcut menu. Click **OK** to submit the changes.

2. The dialog box in *Figure 5.27* opens and displays the reaction to which the connector is linked.

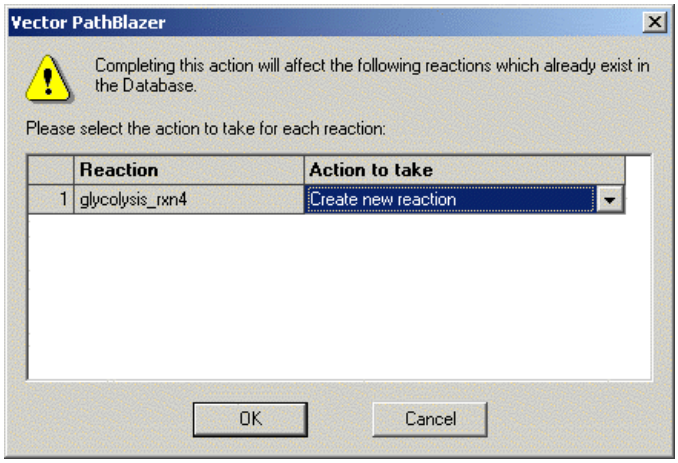


Figure 5.27 Dialog box that displays when changing or deleting a connector from a saved reaction

3. Select an option from the drop-down list in the **Action to take** column. See Table *5.3* for action descriptions.

Note: If a reaction participates in more than one pathway, the option **Update reaction** is not available.

4. Click **OK**. The selected action is applied to the reaction.
5. To save the change, see *Saving PathBlazer Components, Reactions and Pathways* on page 46. To cancel the change and revert to the previous pathway, close the pathway without saving it and then reopen it.

Adding Labels

In the Graphics window, the only object for which a name displays is a component. The pathway's name displays in the title bar but reaction and connector names do not display. You can display the name of a reaction or show information about a connector with a label. **Labels** can

be added to a component, reaction node, or connector to display additional information or titles about one of these objects. A label is not a separate object but is linked to the object to which it is associated in a particular pathway and is saved with the pathway.

Create a label—by selecting an object in the Graphics window and selecting **Create Label** from the shortcut menu. An untitled label is placed near the selected object. Name the label and press ENTER. The label displays next to object (*Figure 5.28*). Labels can be moved anywhere in the Graphics window by selecting the label and dragging it to a new position. When selected, a dotted line shows the object to which the label is connected.

Delete a label—by selecting it and pressing the DELETE-key.

Change a label's display properties—by selecting it and selecting **Object Properties** from the shortcut menu or using the graphics buttons in the Graphics toolbar.

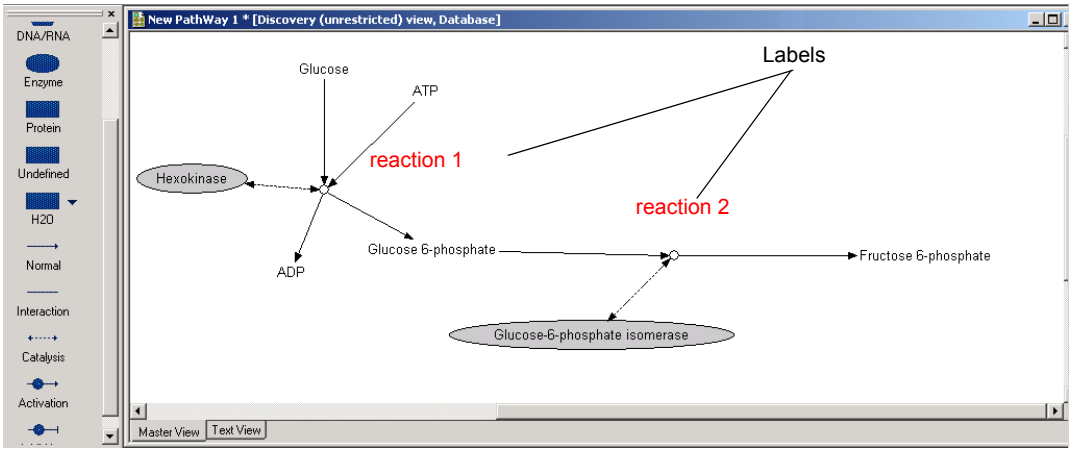


Figure 5.28 Labels added to reaction nodes

CHAPTER

6

AUTOMATICALLY ASSEMBLING PATHWAYS

This chapter describes how to use Vector PathBlazer to suggest novel pathways and protein-protein interaction networks from known components and reactions.

Topics in this chapter include:

- *Introduction* on this page
- *Pathway Assembly Parameters* on page 134
- *Assembling Metabolic Versus Discovery Pathways* on page 138
- *Adding Stepwise Reactions to Pathways* on page 138
- *Building Pathways by Selecting Reactions in the Database Explorer* on page 139
- *Examples of Automatically Assembling Pathways* on page 139

Introduction

The previous chapters described how known pathways are represented in Vector PathBlazer and how to import, draw, and manage known pathways. This chapter describes how to use Vector PathBlazer to perform its most important function: using known pathway and reaction data to build novel pathways.

Many molecules, such as ligands and receptors, are known to participate in many pathways and may effect different reactions under normal and disease states. Suppose you are studying the EGF:EGF receptor interaction in the context of malignant melanoma but you do not know any of the downstream interactions. You want to know, based on the data sets you have loaded into your database (KEGG, BIND, DIP, TransPath, BioCyc, PPI and/or proprietary), what other molecules are known to interact with this complex. To do this you build queries in which you specify a component to build from, to, or through as well as other parameters. Vector PathBlazer then evaluates all the specified reactions and automatically constructs a pathway or network in the Graphics window that includes all possible pathways and interactions that match the query.

Pathway Assembly Parameters

There are two steps in the assembly process. First, you create component and reaction subsets to limit the pathway assembly output. This is key to building a meaningful pathway. Second, you specify the parameters that must be considered when building the pathway.

Specifying Parameters

The **Build a Pathway** dialog box is used to configure a query by which Vector PathBlazer will automatically build a pathway (*Figure 6.1*). To open this dialog box, select **Tools > Build a Pathway** and select either **Build Metabolic Pathway** or **Build Discovery Pathway** from the submenu. The dialog box is the same for Metabolic and Discovery pathways but the results presented in the Graphics window are different.

- When a pathway is built in Discovery View, any enzymes included in the results display as separate components and are pooled. Pooling means that if the enzyme (or other component) is included in more than one reaction in the pathway, it is represented only once in the Graphics window. Connectors are drawn from the single component to any reactions that reference it.
- When a pathway is built in Metabolic View, any enzymes included in the results display as labels of the reaction in which they participate and are not pooled.

The Build a Pathway dialog box consists of several areas with different parameters in each. Each area of the dialog box is described in the following subsections. Following these descriptions, several scenarios are presented for building a pathway using different sets of parameters.

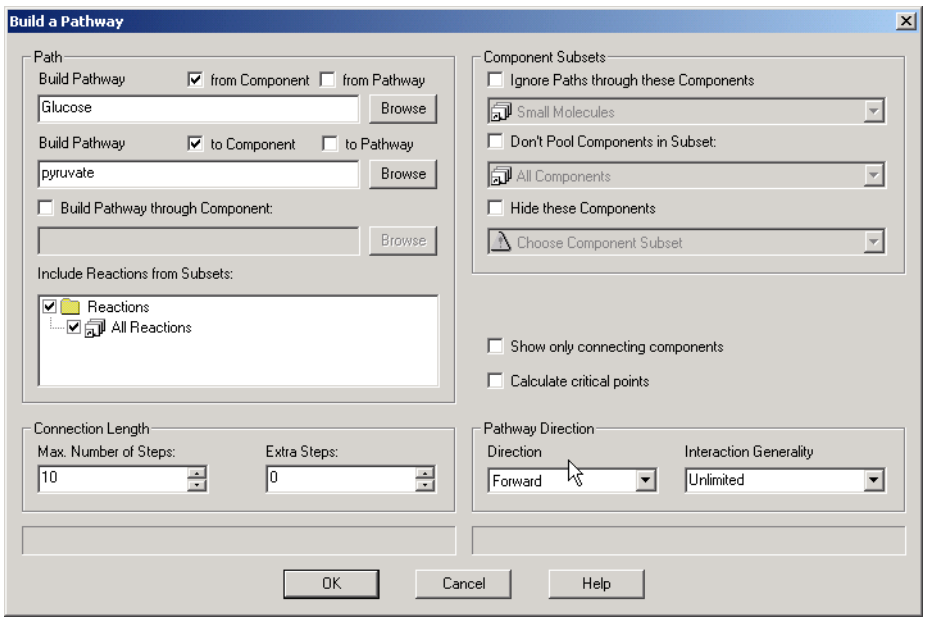


Figure 6.1 Build a pathway dialog box

Selecting Components and Reactions

To create a meaningful pathway, you should create pathway, component and reaction subsets before you configure a query to automatically generate a pathway. Subsetting effectively groups components, pathways, and reactions that are likely to participate in a pathway. Components can then be quickly selected from pre-built subsets for starting and ending the pathway as well as limiting the pathway. For more information about creating subsets, see *Organizing Pathway Data* on page 33 and *Searching Objects in the Database and Creating Subsets* on page 54.

The **Path** box in the upper left of the Build a Pathway dialog box is for specifying start, end, and through components or pathways when building a pathway. Specifying start, end, and through components or pathways is optional. When Vector PathBlazer is given the name of two components or pathways, it generates potential pathways from one component or pathway to the other. If only the start component is specified and a number of steps is defined (see below), the program generates all pathways from the start component or pathway up to *n* number of steps.

Identify the start component or pathway—by selecting the **Build Pathway from Component** or the **Build Pathway from Pathway** checkbox. Selecting a start component or pathway is optional. You can either browse for the starting component or pathway by clicking the **Browse** button and selecting a component or pathway from one of the component subsets in the database or you can enter the name of the component or pathway in the text field. Synonyms can also be entered in the text field. You can build a pathway in either the forward or reverse direction from that component or pathway. Parameters for direction are described in *Specifying Pathway Direction and Interaction Generality* on page 137. For example, if pyruvate is specified as the starting component and glycolysis is built in the reverse direction, the result is a pathway ending in glucose. If glucose is specified as the starting component and a pathway is built in the forward direction, the result is a pathway ending in pyruvate.

Note: If a synonym is used to identify a component and the synonym is associated with more than one component, a list of components associated with the synonym displays. Select one component from the list and click **OK** to continue.

Identify the end component or pathway—by selecting the **Build Pathway to Component** or the **Build Pathway to Pathway** checkbox. Selecting an end component or pathway is optional. Also, if you do not select this checkbox then the through component checkbox is unavailable. You can either browse for the component or pathway name by clicking the **Browse** button and selecting a component or pathway from one of the component or pathway subsets in the database or you can enter the name of the component or pathway in the text field. Synonyms can also be entered in the text field.

Note: If you are building a pathway from a small subset, it is recommended that an end component is not selected.

Identify the through component—by selecting the **Build Pathway through Component** checkbox. Selecting a through component is optional. You can either browse for the component name by clicking the **Browse** button and selecting a component from one of the component subsets in the database or you can enter the name of the component in the text field. Synonyms can also be entered in the text field.

Select a reaction subset—by selecting from the options in the drop-down list under **Include Reactions from Subset**. To use all reactions in the database, select the **All Reactions** subset. Only reactions present in the selected subset are considered during the assembly. For example, if a reaction subset is selected that contains reactions that only occur in the human, only those reactions (that is, valid in human) are used in assembling the pathway.

Using Component Subsets to Limit Pathway Interactions

The **Component Subset** box in the upper right of the Build a Pathway dialog is for specifying which components you want to exclude, pool, or hide when assembling pathways.

In addition to specifying the start and end components, you can identify component subsets to be excluded from the pathway during assembly. This is an effective way to reduce the number of reactions in the display. For example, if you create a subset of common components, such as water, ATP, UTP, etc, you can specify that these should be excluded from the pathway assembly and the pathway will be constructed without creating paths through these components. Excluded components, if they exist in reactions, are shown in the pathway, but are not used in linking reactions together during pathway construction.

Pooling refers to drawing a component that occurs more than once in a pathway one time in the Graphics window with multiple connectors drawn to the reactions in which it is involved. When a pathway is assembled, the default is to pool components that occur more than once. You can select to not pool components, in which case, each occurrence of a component in an assembled pathway is drawn separately. You can also select a subset that contains components you specifically do not want pooled such as small molecules or enzymes.

Note: If a particular component is specified to not be pooled and it is required to build a pathway, the displayed pathway will be disconnected. For example, the pathway A > B > C can be built if B is pooled from the reactions A >B and B > C. If B is not pooled then the reactions A > B and B > C are displayed as disconnected reactions in the pathway even though they share B as a common component.

Hiding components is useful when you want certain components to be used in assembling the reaction but you do not want them displayed.

Exclude components—by selecting the **Ignore Paths through these Components** checkbox and then selecting a component subset from the drop-down list.

Turn off component pooling—by selecting the **Don't Pool Components in Subset** checkbox and then selecting a component subset from the drop-down list.

Hide components—by selecting the **Hide these Components** checkbox and then selecting a component subset from the drop-down list.

You can select all three of the checkboxes or none of the checkboxes depending on how you want to configure these parameters.

Show only connecting components—by checking the **Show Only Connecting Components** checkbox, only connecting components will be displayed in the pathway you are building.

Find components or reactions that disrupt the pathway—by checking the **Calculate Critical Points** checkbox. This set of components and/or reactions will constrict the pathway. When they are deleted, they will disrupt the pathway or increase its length. These constricting elements of the pathway display in a color unique from the other colors in the pathway.

Limiting the Number of Steps Between Components

The **Connection Length** box in the lower left of the Build a Pathway dialog is for specifying the maximum number of steps that can be used to assemble a pathway. The algorithm identifies the shortest possible pathway between two points based on the maximum number of steps entered. If the number of steps of the shortest possible pathway is less than or equal to the maximum number of steps entered, then the pathway is displayed. If the shortest pathway is three steps and you have specified ten steps, then all pathways with a length of three steps are shown. If the shortest pathway has more steps than the specified limit, a message displays that a pathway could not be constructed from the parameters. You can then modify the parameters and attempt to build the pathway again. You can also add a range of steps for consideration. For example, if the shortest possible path is three steps but there is also a pathway of four steps and one additional step is specified then both three and four step pathways are displayed. If two additional steps are specified, then all pathways of lengths three, four, and five steps are displayed.

Set the maximum number of steps—by changing the value in the **Max number of steps** field. For a pathway to be built, this value must be greater than or equal to one and less than or equal to 254.

Specify additional steps—by changing the value in the **Extra Steps** field.

These fields are only available if the **Build Pathway to Component** checkbox is selected in the **Path** box.

Specifying Pathway Direction and Interaction Generality

The **Pathway Direction** box in the lower right of the Build a Pathway dialog is for specifying pathway direction and interaction generality. Pathway direction refers to the direction in which connectors are followed during pathway assembly. Connector direction does not necessarily refer to the direction a chemical reaction proceeds biologically.

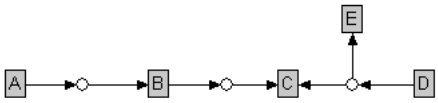
If direction should be considered in pathway assembly, there are three options:

Forward: the pathway backbone is constructed by following connectors in reactions in a forward direction or in the direction connectors point. For example, in the biological sense as well as in the way it is represented in Vector PathBlazer, glycolysis proceeds from glucose to pyruvate in a series of steps and the connectors in the reaction are represented in a left to right pointing direction: glucose -> glucose-6-phosphate, etc. If this pathway were built in a forward direction, the program would follow the connectors in each reaction from glucose to pyruvate.

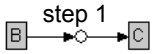
Backward: the pathway backbone is constructed by following connectors in a backward direction or against the direction connectors point. For example, in the biological sense, glycolysis does not run in a backward direction from pyruvate to glucose. However, if the program is instructed to build a pathway in the reverse direction from pyruvate to glucose, it will build against the direction connectors point.

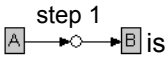
Ignore [direction]: the pathway backbone is constructed without considering direction.

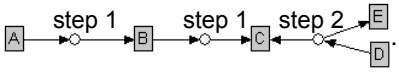
Consider the following examples in terms of the pathway:



If the program is instructed to assemble a pathway from:

- B in two steps in the forward direction then only the one-step pathway  is returned because the program goes in the direction of the connectors from B to C but in the next reactions between C and D and E, the connectors point in the opposite, or backwards, direction so the program does not consider these reactions.

- B in two steps in the backward direction then only the one-step pathway  is returned because the program goes against the direction of the connectors from B to A.

- B in two steps and ignore direction then the following pathway is returned in which directions is not considered  .

Specify the forward direction—by selecting **Forward** from the drop-down list in the **Direction** field.

Specify the backward direction—by selecting **Backward** from the drop-down list.

Specify no direction—by selecting **Ignore** from the drop-down list.

Interaction generality refers to protein-protein interactions. It is defined as the number of proteins that directly interact with the target protein pair minus the number of proteins interacting with more than one protein plus one. In general, the lower the generality score, the more biologically relevant a protein-protein interaction. Protein-protein interactions extending from a specific protein that have an interaction generality score lower than that set will be used in the assembly of the protein-protein interaction network. Interaction generality is undefined for interactions with more than two components.

Build a network of protein-protein interactions—by selecting **Ignore** from the drop-down list.

Set the interaction generality score—by selecting a value from the drop-down list in the **Interaction Generality** field. The default setting is **Unlimited**. If **Unlimited** is selected, all possible interactions, regardless of biological relevance, are shown.

Pathway Colors in the Graphics Window

When you are building a pathway in the Graphics Pane, pathway elements display in the following default colors:

Field	*Description*	*Color*
Through	Component in "the Build pathway through component" field	aqua
Start component	Component from which pathway begins	aqua
End component	Component with which pathway ends	red
Shortest new path	Shortest new path or first path from which a pathway begins	deep aqua
All other new paths	Secondary or paths other than shortest new path	teal to dark blue
Critical points	Critical points in the pathway	dark gray blue
Components not involved in any path	Components that are not *directly* involved in a pathway or reaction, but are involved in a *peripheral* way, such as a catalyst)	no color

Assembling Metabolic Versus Discovery Pathways

When you draw a pathway in Metabolic View in the Graphics window and you want to include a catalyzing enzyme with a connector (represented as a double-headed arrow), the catalyzing reaction is included as a label of the reaction. Similarly, when Vector PathBlazer automatically assembles a reaction in Metabolic View, any catalyzing connector that links to a reaction node is drawn as a label of the reaction. In Discovery View, on the other hand, the connector and the catalyzing agent (that is, the enzyme) are drawn as separate elements in the reaction. The following are the other restrictions that apply when assembling a pathway in Metabolic View.

1. Catalyzing agents are not pooled. If the same enzyme occurs more than once in a pathway, a label displays for each occurrence.
2. Any effecting agents are not drawn to the catalyzing agents.
3. Protein-protein interactions (PPI) are not displayed and pathways are not assembled through PPI interactions. If a PPI reaction is included in a subset of reactions that are used to assemble a pathway, a warning displays and you can select to either proceed without the reaction(s) or stop the assembly.

Adding Stepwise Reactions to Pathways

Once a pathway is generated, you can select a component and ask to see the next level of components connected directly to that specific component (that is, the next reaction). When you select a component, one of the options is "1 more step". The second level from that option allows you to specify whether the next reaction should come from the reaction subset used to

assemble the original pathway (if one was used) or from the database of all reactions. You can also specify whether you want directed reactions (that is, the next level is from reactions that have a direction associated with them in the reaction, such as metabolic or signal transduction reactions) or non-directed reactions (that is, protein-protein interactions). The interaction generality score for protein-protein interactions (if “non directed components” is chosen) is set to the level used to generate the original pathway. If no interaction generality score was set for the original pathway, the default value is infinite, thereby showing all possible interactions.

Building Pathways by Selecting Reactions in the Database Explorer

Pathways can also be built by selecting two or more reactions in the Database Explorer. When pathways are built this way, the program attempts to link common components from the selected reactions into a network or pathway. Discovery View is the default display view and components are pooled. In addition, there is no way to select a from or to component.

To build a pathway from reactions selected in Database Explorer, use the following steps.

1. Select two or more reactions in the List Pane and select **Build a Pathway** from the shortcut menu.
2. The resulting network or pathway displays in the Graphics window.

Examples of Automatically Assembling Pathways

The following examples show the pathways that are automatically assembled when certain parameters are selected. Each example is illustrated using components and reactions from the metabolic pathway glycolysis to show how the algorithm assembles a pathway in the context of a well known pathway. The reactions include the steps of glycolysis and all the components involved in the reactions including the small molecules like ATP, etc. Each example shows the input components and parameters, the expected output, instructions for assembling the pathway, and how the assembled pathway displays in the Graphics window.

Before You Begin

Create a reaction subset that contains all of the reactions in the glycolysis pathway. In the All Pathways folder in the Database Explorer, right click on a glycolysis pathway and select **Create Reaction Subset**. Name the subset **Glycolysis** and click **Create**.

Each of the examples in this section uses a filter applied to a small molecule subset that includes H_20, ADP, and ATP. This **Small Molecule** subset loads with the PathBlazer demo database.

Building a Pathway from a Starting Component

Description This example describes how to assemble a pathway by entering a “from” component only.

Input Starting component is Glucose; number of pathway steps is three

Output Reaction 1: Glucose + ATP + hexokinase -> glucose 6-phosphate + ADP

Reaction 2: glucose-6-phosphate + glucose phosphate isomerase -> fructose-6-phosphate

Reaction 3: fructose-6-phosphate + ATP + phosphofructokinase -> fructose 1,6-bisphosphate + ADP

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**.
2. In the Build a Pathway dialog box, select the **Build Pathway from Component** checkbox and enter **Glucose** as the starting component.

3. Uncheck the **Build Pathway to Component** checkbox.
4. In the **Include Reactions from Subset** field, check the checkbox for the **Glycolysis** reaction subset.
5. Set **Max number of steps** to 3.
6. Select **Ignore Paths through these Components** and select the **Small Molecules** subset. containing the small molecules ATP, ADP, and H_2O.
7. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
8. Set **Direction** to **Forward**.
9. Set **Interaction Generality** to **Unlimited**. The Build a Pathway dialog box should look similar to that in *Figure 6.2*.

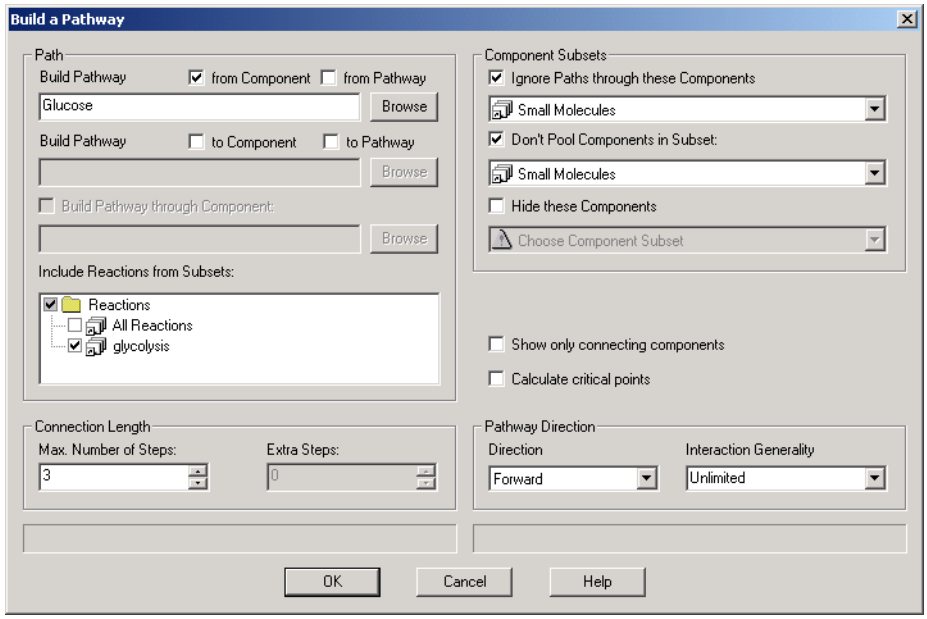


Figure 6.2 Building a pathway from a selected component

10. Click **OK** to start assembling the pathway. A progress bar at the bottom of the window shows the status of the assembly. A dialog box displays informing you of the shortest pathway and the total number of reactions that will display. As expected, the shortest path between glucose and fructose-6-phosphate occurs in three reactions. Click **Yes** to continue.
11. After the assembled pathway displays, save it in the database. Select **File > Save As**. In the Save: Pathway dialog box, in the **Select a Subset** field, select **Metabolic** from the drop-down menu. In the Name text box, enter an appropriate name to identify the pathway, such as **Pathway Glycolysis 2 steps**.

Assembled Pathway

Based on the selected parameters, the program assembles a pathway that consists of the first seven reactions in glycolysis and displays it in the Graphics window (*Figure 6.3*). The starting component is indicated by shading it royal blue. The title bar indicates that the pathway is automatically generated and, when you save the pathway, the name **Automatically generated** is

entered as the default in the Name field. For instructions on saving pathways, see *Saving PathBlazer Components, Reactions and Pathways* on page 46.

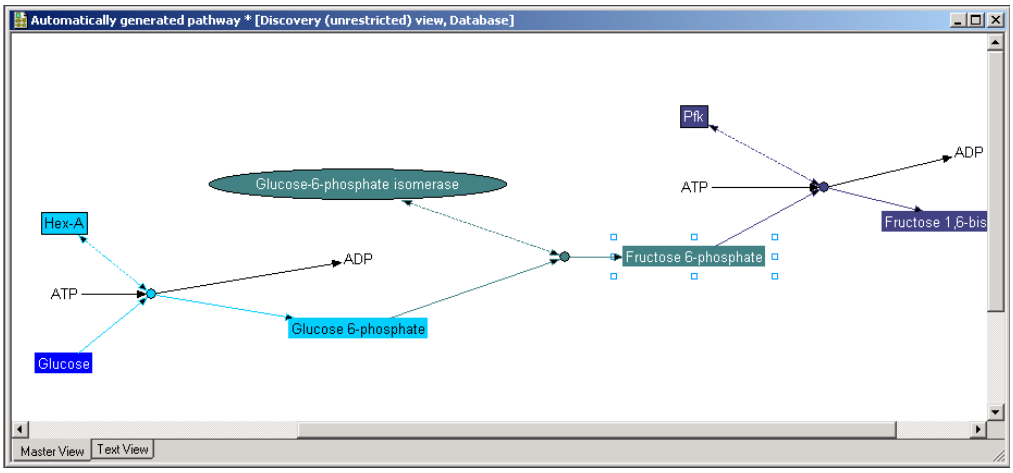


Figure 6.3 First three steps of glycolysis automatically assembled; the font of some molecules has been changed to white

Building a Pathway from a Starting Component to an Ending Component

Description This example describes how to assemble a pathway by entering a “from” component and a “to” component.

Input Starting component is Glucose; ending component is pyruvate; number of pathway steps is nine.

Output Reaction 1: Glucose + ATP + hexokinase -> glucose 6-phosphate + ADP

Reaction 2: glucose-6-phosphate + glucose phosphate isomerase -> fructose-6-phosphate

Reaction 3: fructose-6-phosphate + ATP + phosphofructokinase -> fructose 1,6-bisphosphate + ADP

Reaction 4: fructose 1,6-bisphosphate + fructose diphosphate aldolase-> glyceraldehyde-3-phosphate (2)

Reaction 5: glyceraldehyde-3-phosphate (2) + Pi (2) + NAD^+ (2) +glyceraldehyde phosphate dehydrogenase -> 1,3-bisphosphoglycerate (2) + NADH (2) + H^+ (2)

Reaction 6:1,3-bisphosphoglycerate (2) + ADP (2) + phosphoglycerate kinase -> 3-phosphoglycerate (2) + ATP (2)

Reaction 7: 3-phosphoglycerate (2) + phosphoglyceromutase -> 2-phosphoglycerate (2)

Reaction 8: 2-phosphoglycerate (2) + enolase -> phosphoenolpyruvate (2) + H20 (2)

Reaction 9: phosphoenolpyruvate (2) + ADP (2) + pyruvate kinase -> pyruvate (2) + ATP (2)

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**.
2. In the Build a Pathway dialog box, select the **Build Pathway from Component** checkbox and enter **Glucose** as the starting component.
3. Select the **Build Pathway to Component** checkbox and enter **Pyruvate** as the ending component.
4. In the **Include Reactions from Subset** field, select the **Glycolysis** reaction subset.
5. Set **Max number of steps** to 10.

6. Set **Extra steps** to 0.
7. Select **Ignore Paths through these Components** and select the **Small Molecules** subset.
8. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
9. Set **Direction** to **Forward**.
10. Set **Interaction Generality** to **Unlimited**. The parameters should look similar to those in *Figure 6.4*.

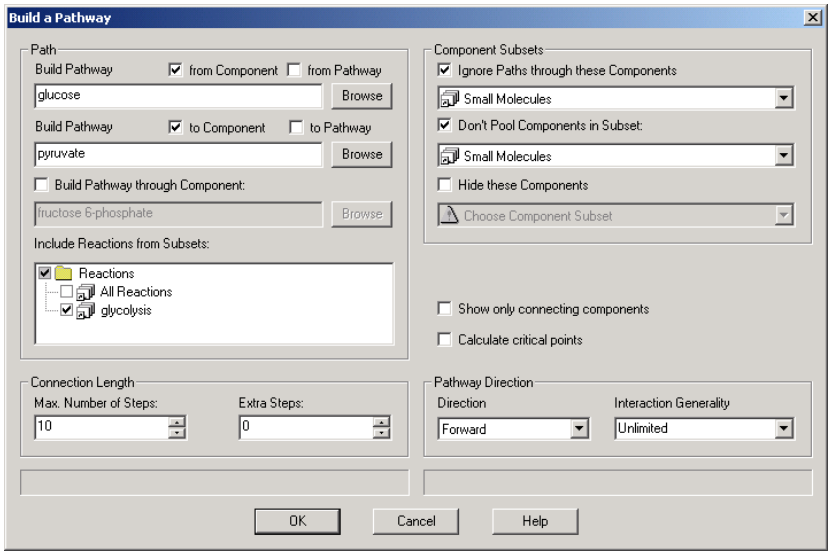


Figure 6.4 Building a pathway from one component to another component

11. Click **OK** to start assembling the pathway. A dialog box displays informing you of the shortest pathway and the total number of reactions that will display. Click **Yes** to continue.

Assembled Pathway

Based on the selected parameters, the program assembles a pathway that consists of the steps in glycolysis. The pathway displays in the Graphics window and is initially enlarged so you can easily view the components (*Figure 6.5*). Use the buttons in the toolbar and the commands in the View menu to resize the image. The starting component is shaded green and the ending component is shaded red (*Figure 6.6*).

The title bar indicates that the pathway is automatically generated and, when you save the pathway, the name **Automatically generated** is entered as the default in the **Name** field.

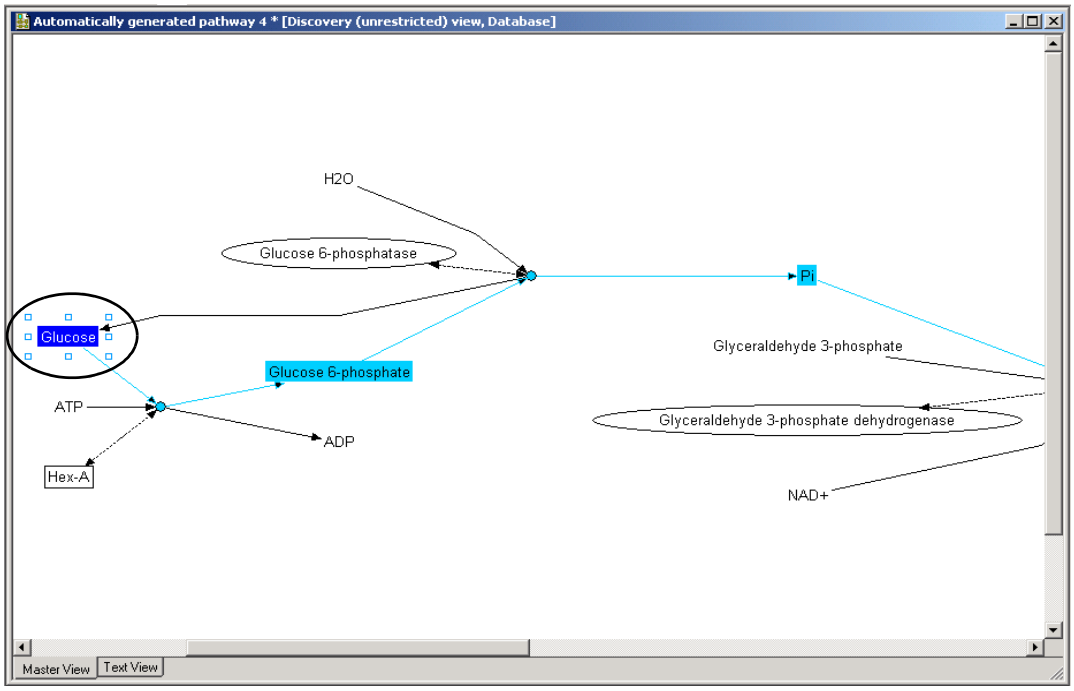


Figure 6.5 Glycolysis automatically assembled with starting component shaded royal blue (font modified to white for image)

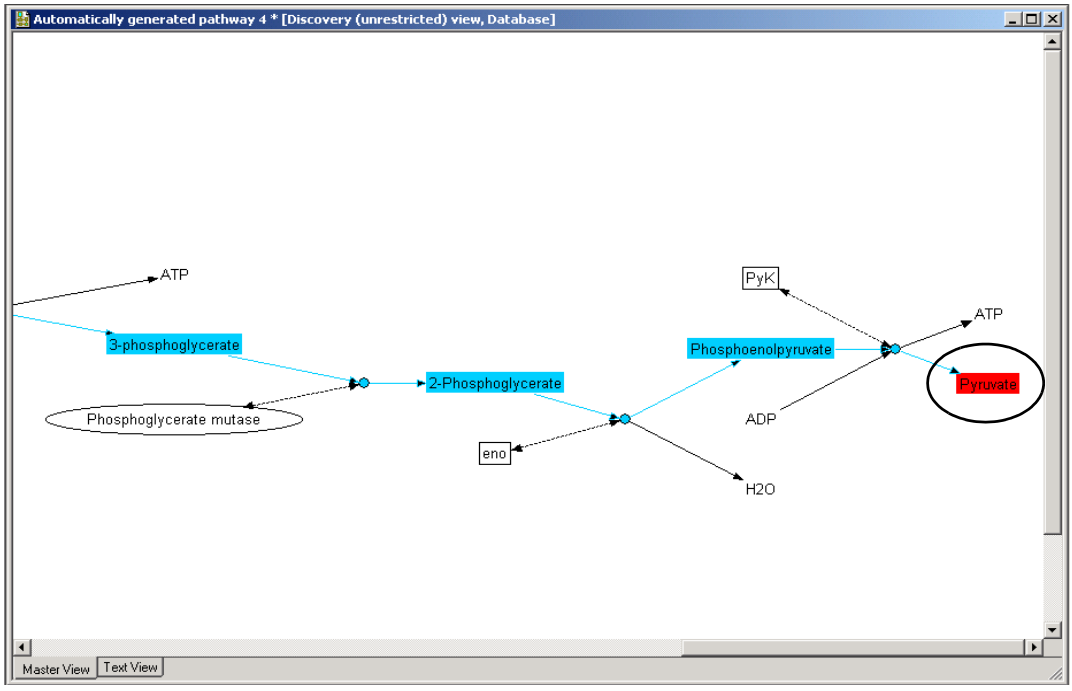


Figure 6.6 Ending component is shaded red

Building a Pathway from a Starting Pathway to an Ending Component

Description This example describes how to assemble a pathway by entering a “from” pathway and a “to” component.

Input The starting "object" is the pathway created and saved in the first example, *Building a Pathway from a Starting Component* on page 139. The ending component is pyruvate; the number of pathway steps is nine.

Output Nine reactions that are the same as for previous example

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**.
2. In the Build a Pathway dialog box, select the **Build Pathway from Pathway** checkbox. Click the **Browse** button and locate the pathway **Pathway glycolysis 2 Steps** (or by the name you assigned the pathway when you created it).
3. Select the **Build Pathway to Component** checkbox and enter **Pyruvate** as the ending component.
4. In the **Include Reactions from Subset** field, select the **Glycolysis** reaction subset.
5. Set **Max number of steps** to 10.
6. Set **Extra steps** to 0.
7. Select **Ignore Paths through these Components** and select the **Small Molecules** subset.
8. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
9. Set **Direction** to **Forward**.
10. Set **Interaction Generality** to **Unlimited**. The parameters should look similar to those in *Figure 6.7*.

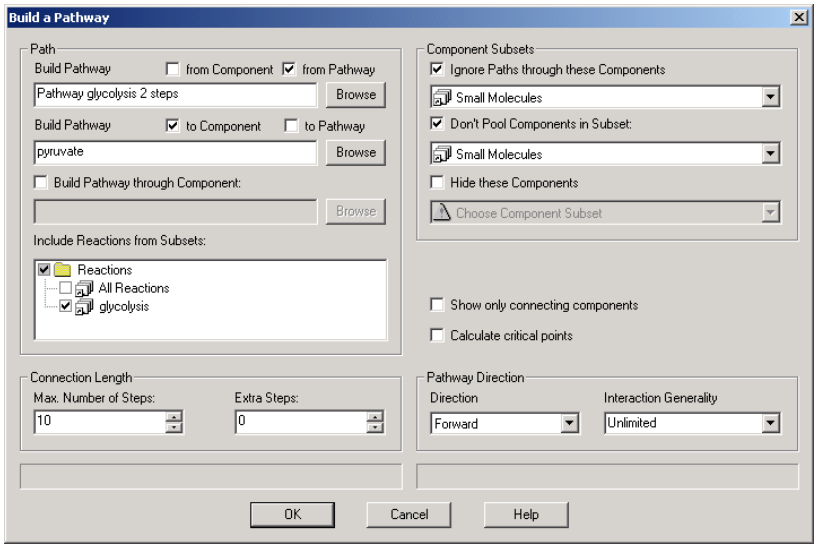


Figure 6.7 Building a pathway from one pathway to another component

11. Click **OK** to start assembling the pathway. A dialog box displays informing you of the shortest pathway and the total number of reactions that will display. Click **Yes** to continue.

Assembled Pathway Based on the selected parameters, the program assembles a pathway that consists of the steps from the first two steps of glycolysis to pyruvate (*Figure 6.5*). The starting pathway is shaded light blue, and the ending component in the starting pathway is shaded royal blue (displayed with white font). The ending component is shaded red (not shown).

The title bar indicates that the pathway is automatically generated and, when you save the pathway, the name **Automatically generated** is entered as the default in the **Name** field.

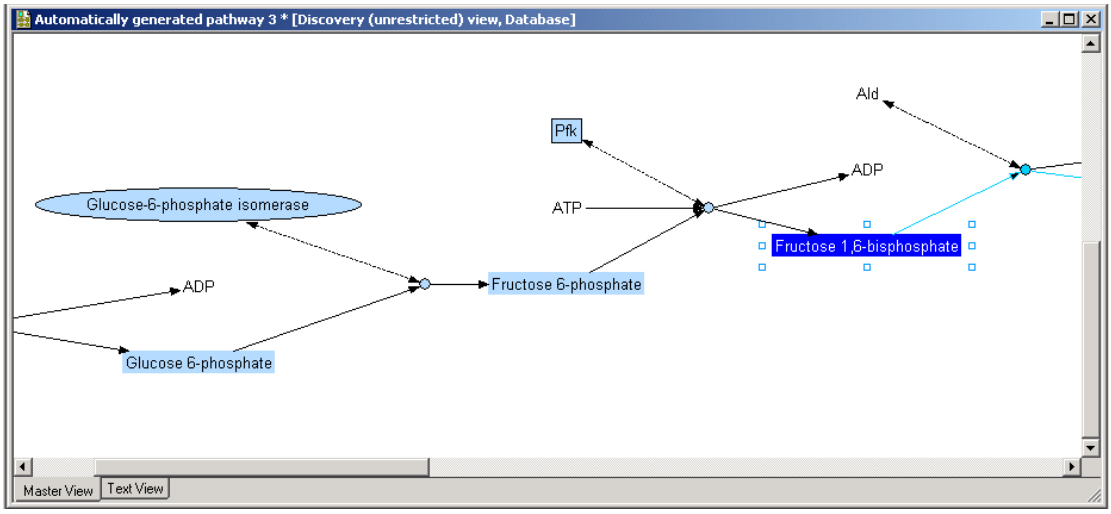


Figure 6.8 Building a pathway from a starting pathway to a component

Building a Pathway Through a Component

Description This example describes how to assemble a pathway by entering a “from” component, a “to” component, and a “through” component.

Input Starting component is Glucose; ending component is Pyruvate; through component is Fructose 6-phosphate; number of pathway steps is ten

Output Same as previous example

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**
2. In the Build a Pathway dialog box, select the **Build Pathway from Component** checkbox and enter **Glucose** as the starting component.
3. Select the **Build Pathway to Component** checkbox and enter **Pyruvate** as the ending component.
4. Select the **Build Pathway through Component** checkbox and enter **Fructose 6-phosphate.**
5. In the **Include Reactions from Subset** field, select the **Glycolysis** reaction subset.
6. Set **Max number of steps** to 10.
7. Set **Extra steps** to 0.
8. Select **Ignore Paths through these Components** and select the **Small Molecules** subset.
9. Select **Don’t Pool Components in Subset** and select the **Small Molecules** subset.
10. Set **Direction** to **Forward**.

11. Set **Interaction Generality** to **Unlimited**. The Build a Pathway dialog box should look similar to that in *Figure 6.9*.

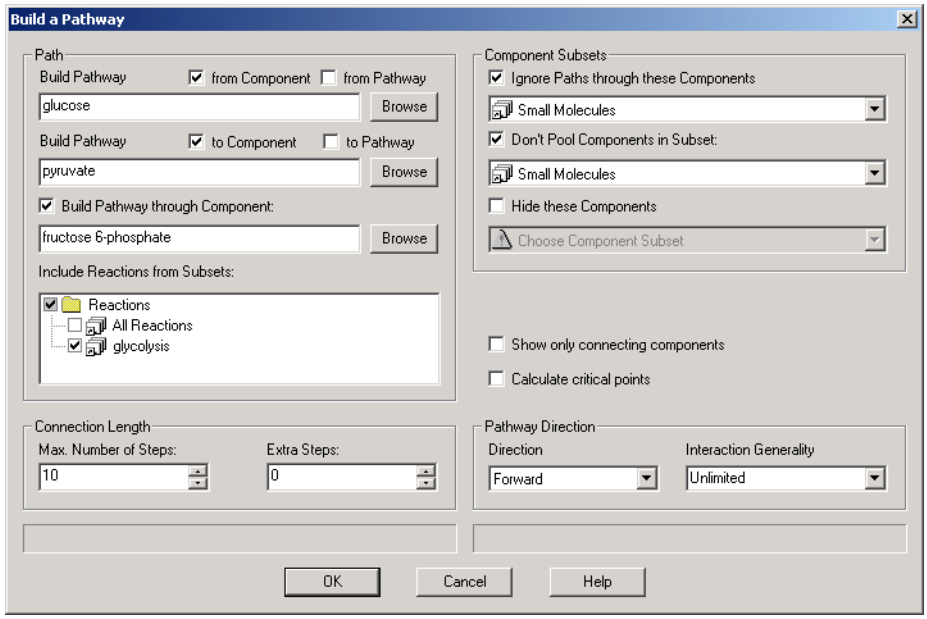


Figure 6.9 Building a pathway through a selected component

12. Click **OK** to start assembling the pathway.

Assembled Pathway

The starting component is royal blue (with font changed to white for this image) and the "through" component is shaded light green (*Figure 6.10*).

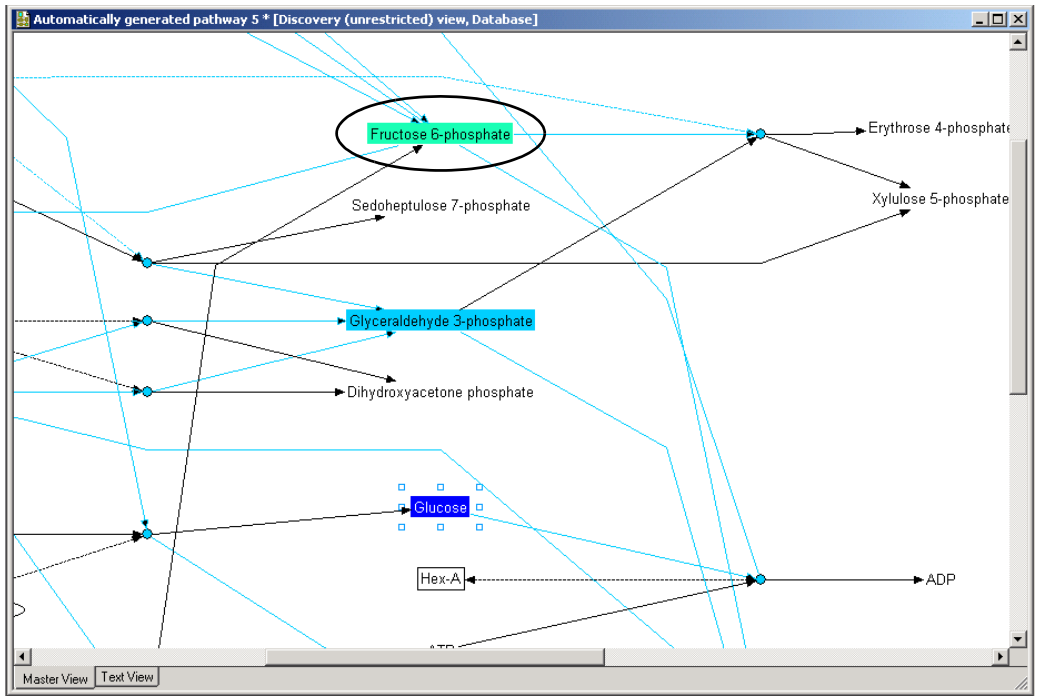


Figure 6.10 Glycoloysis pathway through Fructose 6-phosphate automatically assembled ; the "through" component, Fructose 6-phosphate, is circled

Adding a Stepwise Reaction

Description This example describes how you can add one or more reactions that you specify to an assembled pathway.

Input Starting component is Glucose; number of pathway steps is three

Output Reaction 1: Glucose + ATP + hexokinase -> glucose 6-phosphate + ADP

Reaction 2: glucose-6-phosphate + glucose phosphate isomerase -> fructose-6-phosphate

Reaction 3: fructose-6-phosphate + ATP + phosphofructokinase -> fructose 1,6-bisphosphate + ADP

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**
2. In the Build a Pathway dialog box, select the **Build Pathway from Component** checkbox and enter **Glucose** as the starting component.
3. Select the **Build Pathway to Component** checkbox and enter **Fructose 6-phosphate**.
4. In the **Include Reactions from Subset** field, select the **Glycolysis** reaction subset.
5. Set **Max number of steps** to **3**.
6. Select **Ignore Paths through these Components** and select the **Small Molecules** subset containing the small molecules ATP, ADP, and H_2O.
7. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
8. Set **Direction** to Forward.
9. Set **Interaction Generality** to Unlimited. The Build a Pathway dialog box should look like that in *Figure 6.2*.
10. Click **OK** to start assembling the pathway.
11. The assembled pathway that displays in the Graphics window consists of the first three steps of glycolysis (*Figure 6.11*).

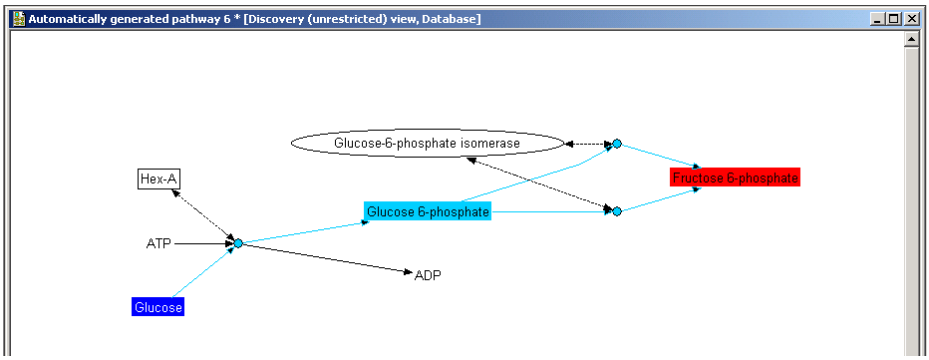


Figure 6.11 First three steps of glycolysis automatically assembled

12. Once the pathway is assembled, the next reaction can be added by searching either the database for all reactions or a reaction subset for reactions including Fructose-6-phosphate. In the Graphics window, right-click on **Fructose 6-phosphate** and select **Add reaction** from the shortcut menu. In the **Add reaction** dialog box, in the drop-down menu, select

Input and select the reaction subset containing the glycolysis reaction entries (*Figure 6.12*). Click **Search**.

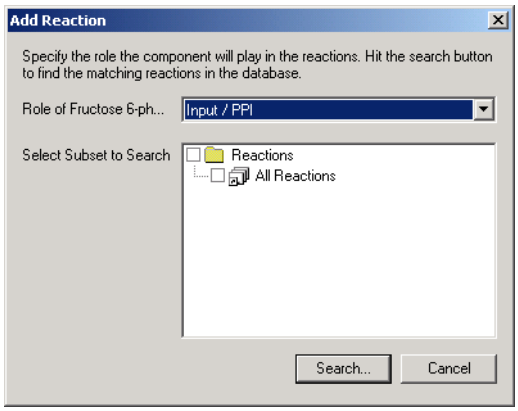

Add Reaction
Specify the role the component will play in the reactions. Hit the search button to find the matching reactions in the database.
Role of Fructose 6-ph...
Input / PPI
Select Subset to Search
Reactions
All Reactions
Search...
Cancel

Figure 6.12 Add reaction dialog box

13. The **Add reaction** dialog box displays all reactions (in this example, there are four) in the selected subset that includes Fructose-6-Phosphate (*Figure 6.13*). Select the reaction(s) and click **OK** to add the reactions to the selected component.

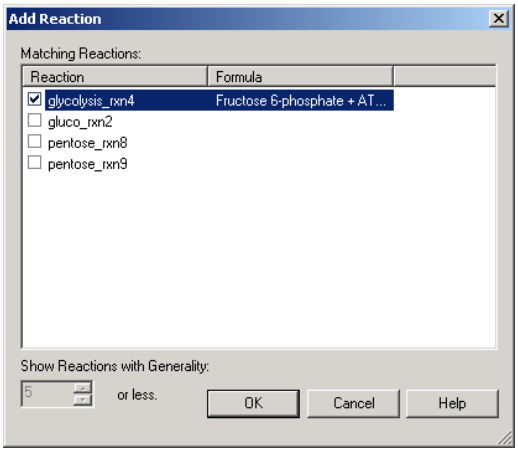


Figure 6.13 Selecting reactions to add to the assembled pathway

Assembled Pathway

Once the reaction is added, the assembled pathway includes the fourth reaction of glycolysis (*Figure 6.14*).

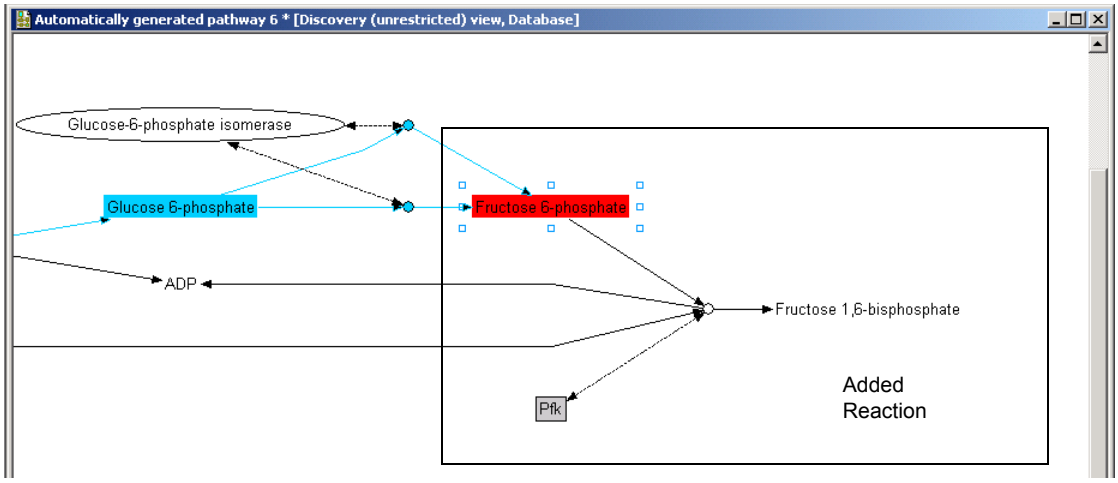


Figure 6.14 Adding a stepwise reaction from a component in an automatically assembled pathway

Note: In *Figure 6.14*, the components ADP and ATP are pooled; that is, instead of redisplaying these components again in the added reaction, the existing components used in the first reaction are reused in the fourth reaction.

Building A Link Between Two Pathways

Description This example describes how to establish a link between two existing pathways.

Input The starting pathway is the glycolysis pathway you built and saved in the first example, *Building a Pathway from a Starting Component* on page 139. The target pathway for the link is the TNFR Signaling Pathway in the PathBlazer database.

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway.**
2. In the Build a Pathway dialog box, select the **Build Pathway from Pathway** checkbox. Click the **Browse** button and locate the pathway **Pathway glycolysis 2 Steps** (or by the name you assigned the pathway when you created it).
3. Select the second **Build Pathway to Pathway** checkbox. Click the **Browse** button and locate the pathway **TNFR Signaling Pathway** in the **Signal Transduction** Pathways subset.
4. In the **Include Reactions from Subset** field, select the **All Reactions** subset.
5. Set **Max number of steps** to 10.
6. Set **Extra steps** to 0.
7. Select **Ignore Paths through these Components** and select the **Small Molecules** subset.
8. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
9. Select the **All Reactions** subset.
10. Set **Direction** to **Ignore**.
11. Set **Interaction Generality** to **Unlimited**.

The Build a Pathway dialog box should look similar to that in *Figure 6.9*.

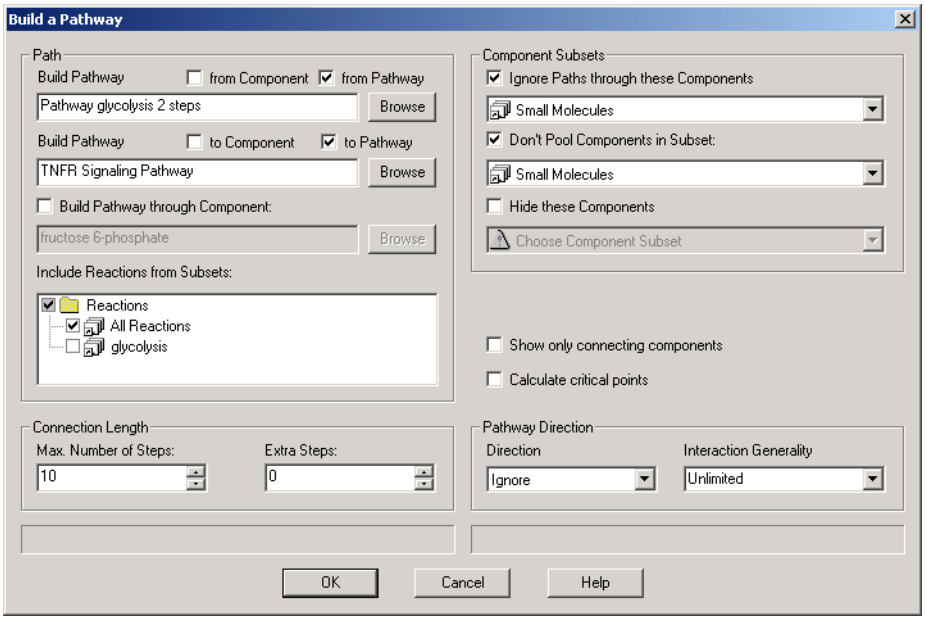


Figure 6.15 Building a link between two pathways

12. Click **OK** to start assembling the pathway.

Assembled Pathway The result displays a complex pathway starting with the Glycolysis 2 Steps pathway and a three step link proceeding to the TNFR Signaling Pathway (*Figure 6.16*). The beginning pathway is shaded light blue; the ending pathway is shaded pink. The start component for the link is royal blue; four end components are shaded red. The links between the two pathways are aqua blue.

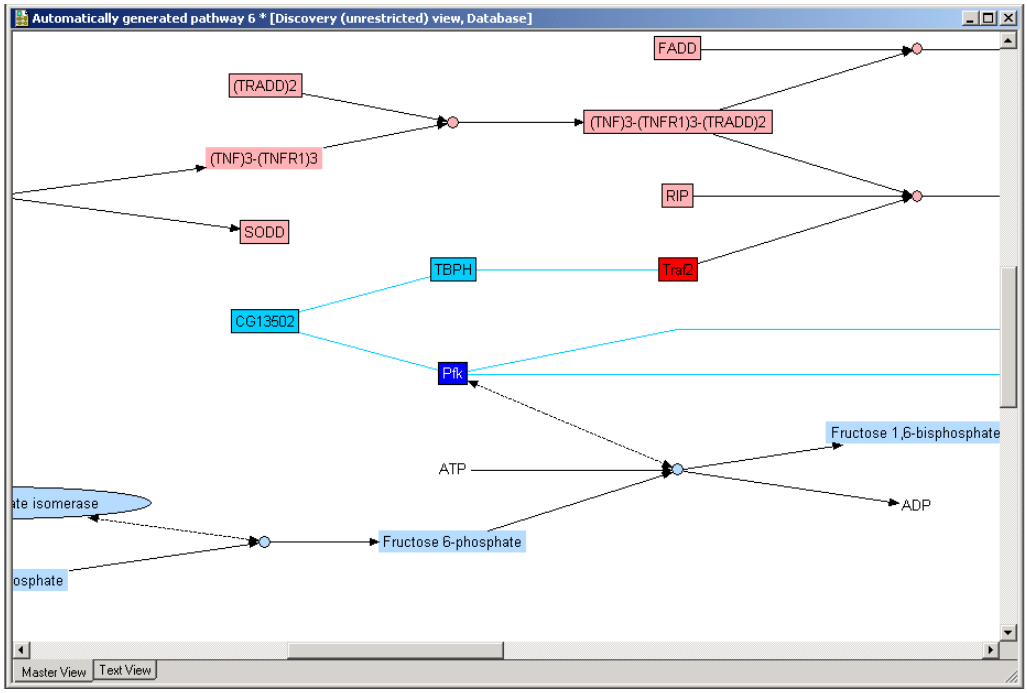


Figure 6.16 Glycolysis pathway linked to the TNFR signaling pathway

Showing Connections to Data from Other Datasources

Description In some instances, you may want to build a pathway from a specific set of reactions and then continue building the pathway by adding data from other reaction subsets or datasources. This example describes how to assemble glycolysis and then add additional reactions that involve hexokinase.

Input Starting component is Glucose; ending component is pyruvate; number of pathway steps is ten.

Output Same as the output in *Building a Pathway from a Starting Component to an Ending Component*.

Steps

1. Select **Tools > Build a Pathway > Build Discovery Pathway**
2. In the Build a Pathway dialog box, select the **Build Pathway from Component** checkbox and enter **Glucose** as the starting component
3. Select the **Build Pathway to Component** checkbox and enter **Pyruvate** as the ending component
4. In the **Include Reactions from Subset** field, select the **Glycolysis** reaction subset.
5. Set **Max number of steps** to 10
6. Set **Extra steps** to 0
7. Select **Ignore Paths through these Components** and select the **Small Molecules** subset.
8. Select **Don't Pool Components in Subset** and select the **Small Molecules** subset.
9. Set **Direction** to Forward

10. Set **Interaction Generality** to Unlimited. The Build a Pathway dialog should look similar to that in *Figure 6.9*.
11. Click **OK** to start assembling the pathway.
12. Once the pathway is assembled, right-click on **Hex-A** (hexokinase) in the Graphics window and select **Add reaction** from the shortcut menu.
13. In the **Add reaction** dialog box, select **Output/PPI** and the reaction subset **All Reactions** to search the entire database. Click **Search**.
14. The Add Reaction dialog box displays all reactions in which Hex-A is included (*Figure 6.17*). Select one or more reactions by selecting the checkbox next to a reaction and click **OK**.

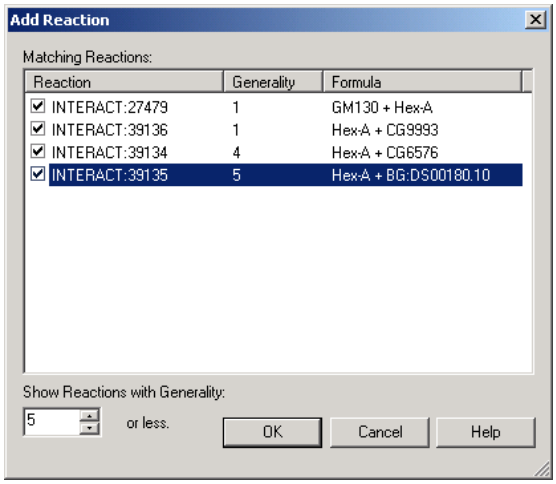


Figure 6.17 Add Reaction dialog box lists all reaction in which selected component is included

Assembled Pathway

The selected reaction is added to hexokinase (*Figure 6.18*). Any components in the added reaction that are already displayed in the pathway are pooled when the reaction is added.

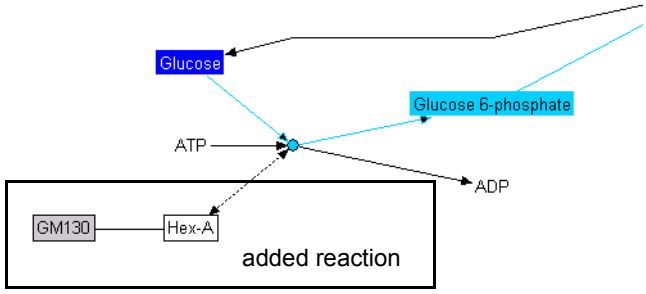


Figure 6.18 Adding reactions to Hex-A (hexokinase)

CHAPTER

7

GENE ONTOLOGIES

This chapter describes gene ontologies, their import and assignment to PathBlazer components, reactions and pathways.

Topics in this chapter include:

- *Introduction to Gene Ontologies* on this page
- *Importing Gene Ontology Terms* on page 154
- *Searching Gene Ontology Terms* on page 156
- *Manual Annotation of PathBlazer Objects with GO Terms* on page 157
- *Importing Gene Ontology Annotations* on page 159
- *Population of Organism/Subcellular Location Attributes Based on GO Annotations* on page 161

Introduction to Gene Ontologies

Gene ontology (GO) is a fixed vocabulary of biological terms that also includes their biological classification(s). Because they are standarized, when gene ontologies are assigned to biological objects, there are no ambiguities in their definitions and classifications.

The Gene Ontology consortium provides two types of information on their website, http://www.geneontology.org,1) the Gene Ontology itself, a fixed vocabulary (dictionary) of terms and their place in classification, (called GO terms file in this chapter) and 2) Gene Ontology annotations, a file of specific GO terms that are already linked to 'real life' or 'common' biological notions such as gene names, biological processes, cell components, etc, (called GO Annotations file in this chapter). For example, the common biological term 'apoptosis' corresponds to gene ontology term GO:0006915.

This chapter is divided loosely into two sections, 1) how to import and use the Gene Ontology terms file, and 2) how to import and use the Gene Ontology Annotations file.

Before you can import any gene ontology files, you must download them to a local directory.

- **File #1**: Gene Ontology dictionary of terms. Download to a local directory the most recent .xml file from http://www.godatabase.org/dev/database/archive/latest/. See *Introduction to Gene Ontologies* in the following section.
- **File #2**: Gene ontology annotations file containing GO terms that are already mapped to genes in a given organism. Select and download to a local directory the specific annotation file you want to use from http://www.geneontology.org/GO.current.annotations.shtml. **Note:** To import this file, you MUST import File #1, GO terms first. Then see *Importing Gene Ontology Annotations* on page 159.

Note: After you have downloaded gene ontology files from the Consortium website, you will need to return to the GO website for periodic updates to your ontologies. See *Updating GO Categories* on page 159.

As the gene ontology *annotations* dictionary is imported, annotations can be assigned to PathBlazer objects. Alternatively, from within PathBlazer, you can manually assign specific gene ontology terms to individual PathBlazer components, reactions and pathways, or in some instances, you can annotate objects in batches. You can group together in a subset objects with a particular ontology classification. You can also perform database searches for objects annotated with specific GO terms.

The GO annotations display on the Properties/GO Annotations tab for Components, Reactions or Pathways. For more information, see *Annotation Fields for Components, Reactions, and Pathways* on page 39.

Working with Gene Ontology Terms

Importing Gene Ontology Terms

After you have downloaded the gene ontology terms dictionary from the Gene Ontology Consortium website (see previous section), you can import the file into PathBlazer. Use the following steps to perform the import:

1. Open PathBlazer.
2. Before launching import, close all PathBlazer display windows.
3. Select **Tools > Manage Gene Ontology > Import Terms**.

4. In the first Gene Ontology Import dialog box that opens, click the **Browse** button to locate and select the file you want to import (*Figure 7.1*). For GO terms, the file must have an .xml extension.

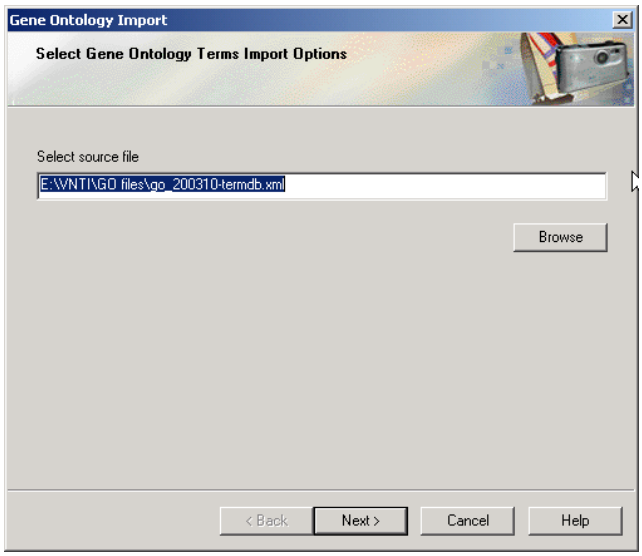


Figure 7.1 Gene Ontology Import Terms dialog box

5. Click **Next**.

At this point, the GO terms will be loaded. The screen displays a monitor showing the import progress. After this import procedure, you can proceed with importing the gene ontology annotation file. See *Importing Gene Ontology Annotations* on page 159.

You can view the GO terms or search for specific terms, assign GO annotations manually to PathBlazer objects, as well as update the gene ontologies at a later point. All of these topics are covered in the following sections.

Viewing Gene Ontology Terms

To view gene ontology terms in PathBlazer, select **Tools > Manage Gene Ontologies > View Gene Ontology Categories**.

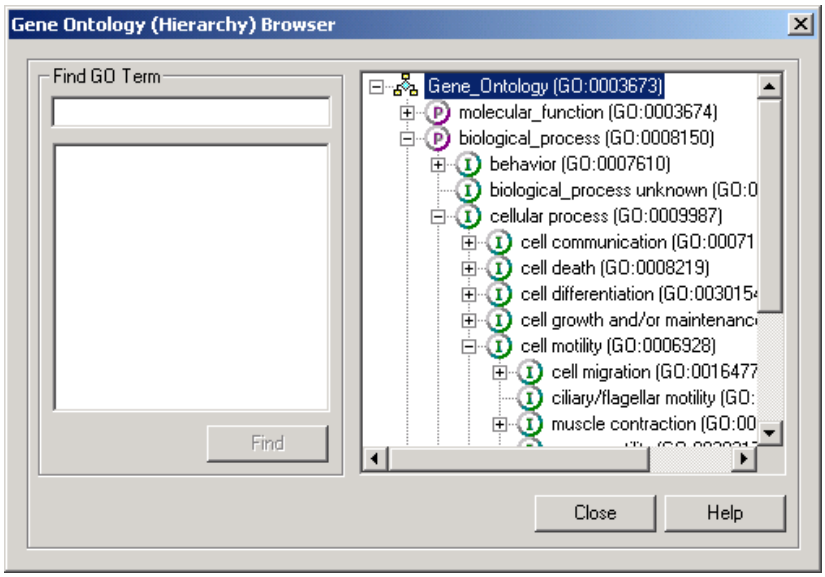


Figure 7.2 Gene Ontology Browser

The Gene Ontology Browser dialog box that opens displays the hierarchical relationships between the gene ontology terms (*Figure 7.2*). The right panel of the dialog box allows you to browse the Gene Ontology "tree" displayed in the right panel. The left panel is used for retrieving gene ontology terms in a search.

Searching Gene Ontology Terms

If you do not know a Gene Ontology term, you can use search capabilities of the GO viewer. Select **Tools > Manage Gene Ontologies > View Gene Ontology Categories**. Enter the query ontology term you want searched in the **Find GO Term** text box, and click the **Find** button (*Figure 7.3*). Search results, as well as the number of terms found display in the left panel. If you click on a line in the results list, the term is highlighted simultaneously in the GO tree in the right panel.

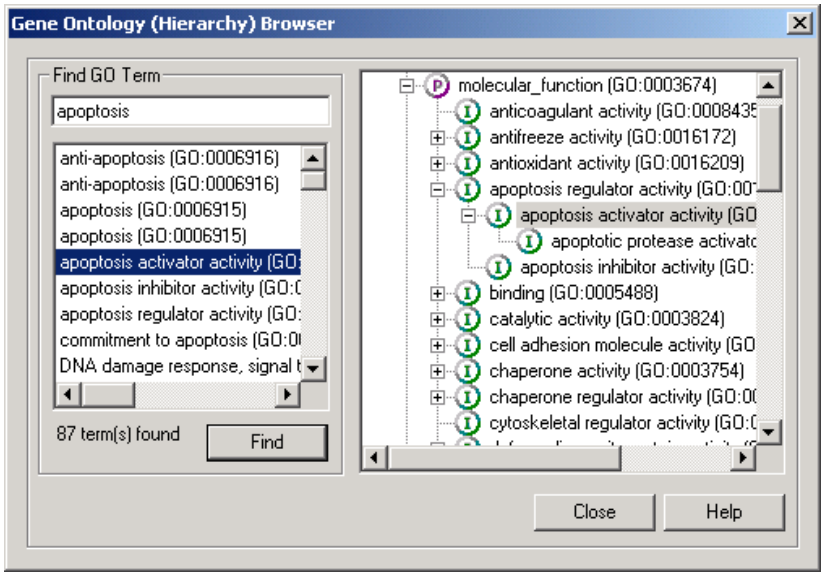


Figure 7.3 Gene Ontology Browser displaying GO term search results

The icons in the displayed tree, borrowed from standard GO viewers, indicate the relationship in the GO tree.

	= 'is a'
	= 'part of'

A child term can be a subclass of ('is a') or a 'part of' its parent. For example, the child **GOterm3** may be a subclass ('is a') of its parent GO term1 and 'a part' *of* its other parent, **GOterm2**. Note that 'part of' means ***can be*** *a part of*, ***not is always*** *a part of*. In other words, the parent need not always encompass the child. For example, in the component ontology, replication fork is a part of the nucleoplasm; however, it is only a part of the nucleoplasm at particular times during the cell cycle.

Alternatively, from the Gene Ontology Browser dialog box, you can link to websites with standard GO viewers. Right click on any gene ontology term in the right pane, and click on one of the links to external viewers (also listed below) (*Figure 7.4*).

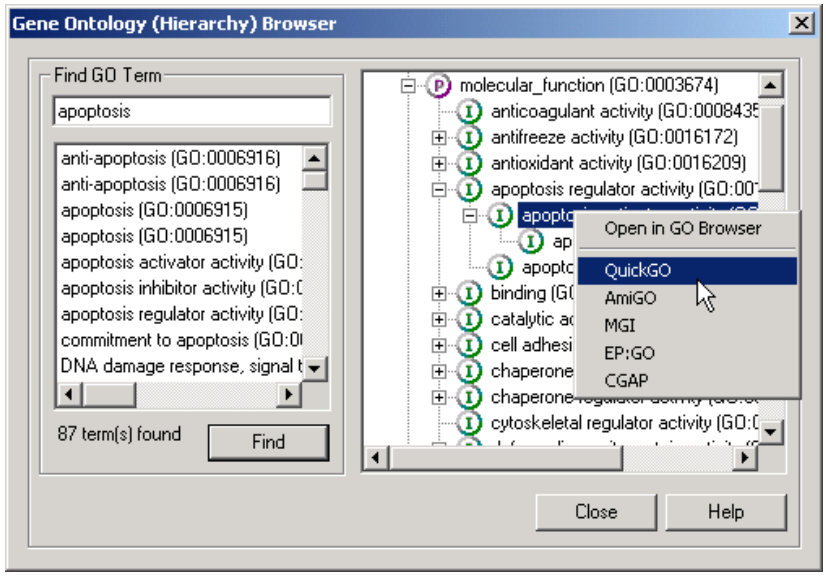


Figure 7.4 Linking to external websites from the GO Browser

- QuickGO
- AmiGO
- MGI
- EP-GO
- CGAP

Their description is located at the following link: http://www.geneontology.org/GO.tools.html.

Note: This dialog box is similar to that used for manually adding and editing gene ontology terms, as described in the following section.

Manual Annotation of PathBlazer Objects with GO Terms

To view, edit or assign GO terms to database objects manually, right click on the component, reaction or pathway in the Explorer List Pane or displayed in a Discovery Pathway Graphics Pane. Select **<object type> Properties** in the shortcut menu. In the <object type> Properties dialog box that opens, select the **GO Annotations** tab.

GO annotations display as a hierarchy on the tab if they have already been assigned to the object from which the Properties dialog box was opened (*Figure 7.5*).

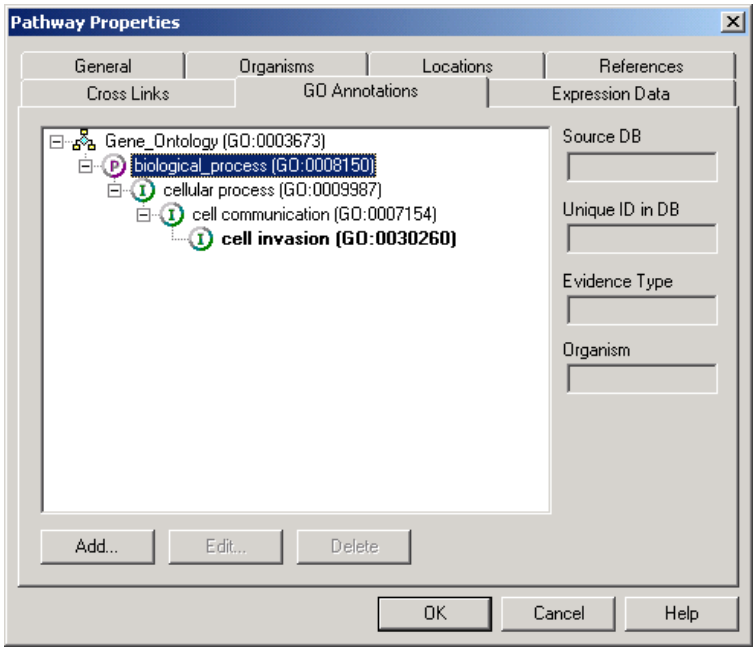


Figure 7.5 Gene Annotations assigned to an object display on the GO Annotation tab of the Properties dialog box

Add a gene ontology annotation—by clicking the **Add** button. In the Add Condition dialog box, select a **GO term** in the right panel (*Figure 7.6*) or launch a search for a gene ontology term of interest by entering the term in the **Find GO Term** text box. Select the term in the results panel; this simultaneously selects it in the right panel.

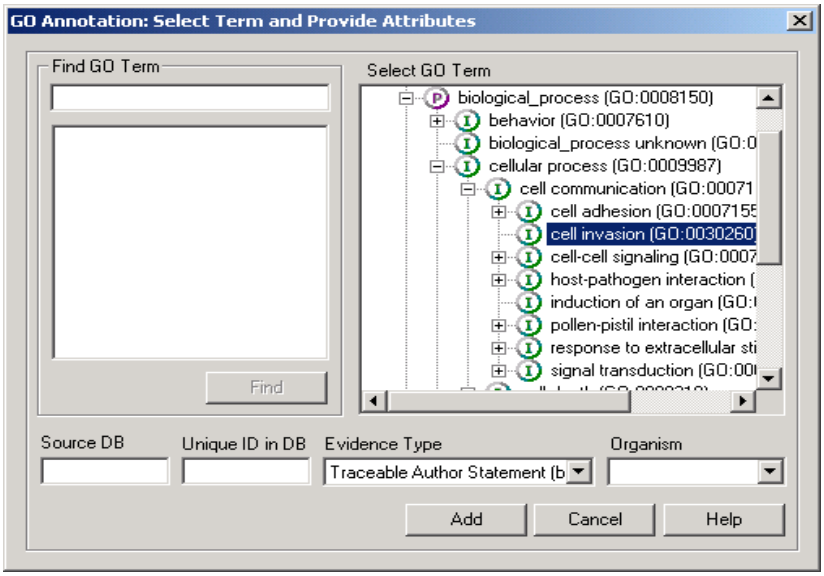


Figure 7.6 In the GO Browser, click on a GO Annotation to assign to an object and click Add.

Click the **Add** button, and the selected annotation is loaded into the GO Annotations tab of the Properties dialog box.

The other two buttons on the GO Annotations tab (**Edit**, **Delete**) do not become available until you select the bottom "leaf" in the tree.

Edit a gene ontology annotation—by selecting the bottom leaf on the annotation tree on the GO Annotations tab and clicking the **Edit** button. In the Edit dialog box that opens (*Figure 7.7*), change current information or add new information (organisms are listed in their order of most frequent usage, highest to lowest). Once you click **OK**, the fields display in the appropriate text boxes on the GO Annotations tab.

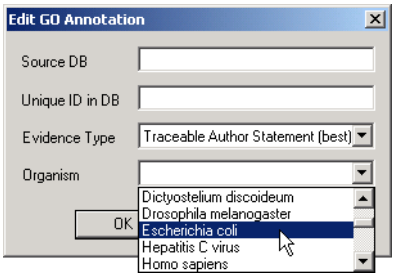


Figure 7.7 Edit GO Annotation dialog box

Delete a gene ontology annotation—by selecting the bottom leaf on the annotation tree on the GO Annotations tab, and clicking **Delete**. The annotation is removed from the object but not deleted from the imported GO Annotations file.

Each GO annotation can have the following attributes:

- **Source Database**--the database from which the GO term originates
- **Unique ID in database**--the ID in the original database
- **Evidence type**--the hierarchy of evidence or confidence in the validity of the annotation
- **Taxonomy**--the organism from which the term or annotation originated

Updating GO Categories

Note: After you have downloaded gene ontology files from the Gene Ontology Consortium website and imported them into Vector PathBlazer, you will need to return to the GO website, http://www.geneontology.org for periodic updates to your ontologies.

When GO categories are imported a second time (updated), all obsolete GO terms are removed and new terms are imported. PathBlazer performs a search for GO annotations, and those database objects that it finds that no longer point to a valid Gene Ontology Term are listed in the GO Annotations Consistency Check dialog box that opens automatically, only if there are terms being removed. In such a case, in the dialog box, you must click on each object noted as having missing annotations and edit the annotations, assigning new ones.

Working with Gene Ontology Annotations

Importing Gene Ontology Annotations

Gene Ontology Annotation is a dictionary which links GO categories to gene names, names of pathways, processes, cell components, etc. These annotations can be applied during import to objects that are already stored in PathBlazer database.

Example: The object Topoisomerase in PathBlazer has crosslinks to SwissProt P11387. Term GO:0003916 has a link SwissProt P11387. After the GO Annotations are imported, Topoisomerase in PathBlazer will be annotated with GO term GO:0003916.

Components, reactions and pathways can have multiple GO annotations from each of three GO categories: **Process**, **Component**, and **Function**. Each GO term could be used in many annotations, in other words could be assigned to many objects.

Notes: Before importing the GO Annotations file, you must first download and import the Gene Ontology dictionary of terms. See *Importing Gene Ontology Terms* on page 154.

After you have downloaded the Gene Ontology Annotations dictionary from the Gene Ontology Consortium website (see *Introduction to Gene Ontologies* on page 153), you can import the GO annotations file into PathBlazer.

Because gene ontology annotations are imported and stored in a PathBlazer database, if you import them into one database and later switch to another – you will have to repeat import of the Gene Ontology itself and Gene Ontology annotation file in the new database.

Use these steps to import gene ontology annotations:

1. Open PathBlazer.
2. Select **Tools > Manage Gene Ontology > Import Gene Ontology Annotations**.
3. In the first Gene Ontology Import dialog box that opens, click the **Browse** button to locate and select the GO annotations file you want to import (*Figure 7.8*). Click **Next**.

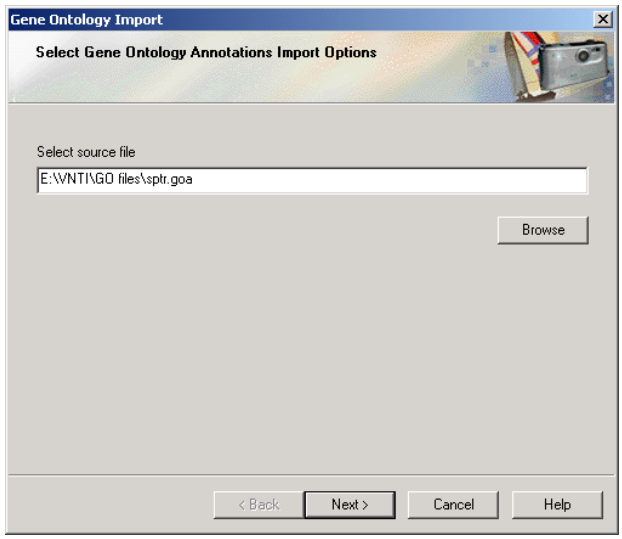


Figure 7.8 Gene Ontology Import Gene Annotations dialog box

4. The second Gene Ontology Import dialog box displays ontology-related database information (*Figure 7.9*). Use this dialog box to map the abbreviations used in GO annotation files to abbreviations present in the current PathBlazer database.

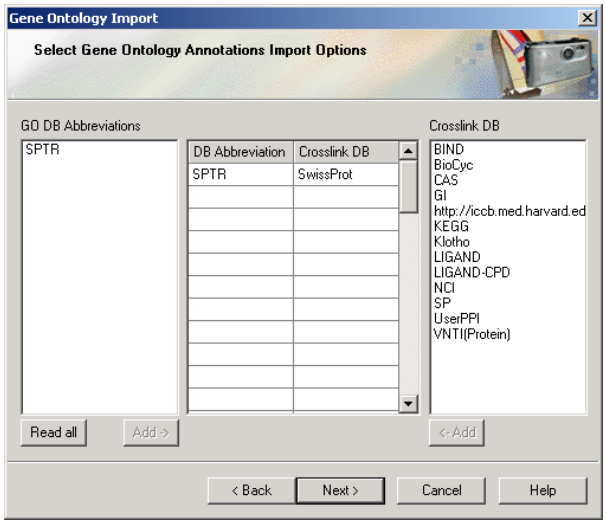


Figure 7.9 Gene Ontology Annotations Import Options

- The left panel, **GO db abbreviations**, displays abbreviations used in annotation files for standard biological databases. For example, SPTR is a frequent abbreviation for Swiss-Prot, but perhaps the abbreviation for Swiss-Prot is different in another database.

  Example: Download the file **sptr.goa** with SwissProt annotations from the Gene Ontology Consortium website. The menu will display the SPTR database abbreviation from the .goa file mapped to the SwissProt link in PathBlazer.

- The right panel of this dialog box, **Crosslink db,** displays abbreviations of databases to which there are cross-links in PathBlazer. The center panel in the dialog box allows you to verify or add abbreviations with their corresponding databases.

To add a term from the left or right panels, select an item and click the **Add** button. Once added to the center column, select and add the matching abbreviation in the opposite panel.

5. Click **Next** to continue. During the import process of gene ontology annotations, the annotations are automatically associated with objects in the database you previously imported into PathBlazer.

To search the database for objects annotated with specific GO annotations, see *Search Database by GO Annotation* on page 61.

Population of Organism/Subcellular Location Attributes Based on GO Annotations

GO annotations can be applied manually, as described on page 157, or they can be applied automatically during the GO annotation import process, described on page 159.

GO annotations store information about the taxonomy (organism) and subcellular location of an object. Using the feature described in this section, you can propagate this information from the GO annotation to the **Organism** and **Subcellular Location** annotation fields of PathBlazer objects. This propagation is assigned to all objects *that already have GO annotations* in a subset you select.

- **Organism Population--**If an object in the PathBlazer database has a GO annotation, and it contains information about taxon (the GO term for organism), PathBlazer adds this organism name to the object's **Organism** annotation field. If before this procedure an object has only a GO annotation, after it, the **Organism** field is also populated.
- **Subcellular Location Population--**If an object in PathBlazer database has a GO annotation, and it contains information about a subcellular location, PathBlazer adds this subcellular name to the object's **Subcellular location** annotation field. Therefore, if before this procedure an object has only a GO annotation, after it, the **Subcellular** location field is also populated.

Use the following steps to assign one of these GO Terms to database objects:

1. Select **Tools > Manage Gene Ontology > Populate <category> Attribute.**

2. In the Populate <category> Attribute dialog box that opens, select **Object type** in the drop-down menu (*Figure 7.10*). Check the checkbox for one or more subsets whose objects will be assigned the annotation, **Subcellular location** or **Organism**.

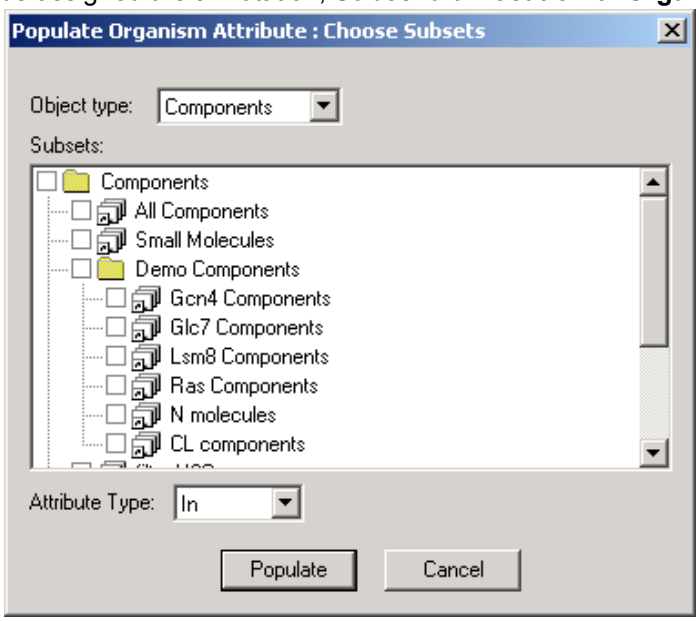


Figure 7.10 Populate <category> Attribute dialog box

3. In the **Attribute Type** drop-down menu, select one of the following logical qualifiers:
   - **In**: Definitively known to be in one or more organisms. If an object is in one or more organisms, all others are excluded.
   - **Known in**: Known to be in an organism but all others cannot be ruled out.
   - **Not in**: Opposite of **Known in**. Known not to be in an organism but all others cannot be ruled out.
4. Click **Populate**. This applies the specified annotations to all objects *that already have GO annotations* in the subset(s).

The newly assigned GO annotations will now appear on the GO Annotations tab in the Properties dialog box for the objects in the selected subset(s).

Sample Workflow Using Gene Annotations

The following example describes a simple workflow using the gene ontology features described in this chapter. Use the default PathBlazer demo db that loads with your PathBlazer 2.0 installation.

1. Download the Gene Ontology Terms file as described in *Importing Gene Ontology Terms* on page 154.
2. Manually annotate 3 components that are part of the glycolysis pathway as described in steps 3-7.
3. From the **All Pathways** folder in the Database Explorer, open the **Glycolysis** discovery pathway.
4. From the Graphics window, for each component listed in the **Component** column of Table *7.1*, open the shortcut menu and select **Component Properties**.
5. On the **Component Crosslinks** tab, click **Add**.

6. In the Crosslink tab, accept **Database** for the **Type** Option. In the **Database** field, enter ***SwissProt***. For each of the components, enter the **Accession ID** displayed in Table *7.1*. This enters links to the SwissProt database for these objects. When you import gene annotations, crosslinks in that file to the SwissProt accession IDs you have just entered for these objects will automatically add GO annotations to these database objects..

Component	*Database*	*Accession ID*
Hexokinase	SwissProt	P19367
Hexokinase	SwissProt	P52789
Phosphofructokinase	SwissProt	P09237
Aldolase	SwissProt	P04075

Table 7.1 Selections for manually assigning GO annotations to selected glycolysis reaction components

7. For each entry, click **OK**, returning you to the Properties dialog box. Note that the GO tab for each is still empty.
8. Import the Gene Annotations file sptr.goa as described in *Importing Gene Ontology Annotations* on page 159. The results should tell you that 27 annotations are imported.
9. After the import, open the Component Properties dialog boxes again to these three objects. Note the GO annotations on the **GO annotations** tab for each. Check the **Organisms** tab for each to see if any objects have current organism annotations. Close the dialog box.
10. Now select **Manage Gene Ontology > Populate Organism Attribute**. In the Populate Organism Attribute dialog box that opens, check the **All Components** subset. In the **Attribute Type**, select **Known In**. This selection means that an object is definitively known to be in certain organisms/locations but it cannot or has not been definitively determined whether it is known to be in other organisms/locations. Click **Populate**. With this feature activated, if there any taxonomy GO annotations for components *that already have GO annotations*, these will be added to the **Organism** tab. Check the **Organism** tabs for the 3 objects again to verify that this operation was executed.

For more information about working with GO Annotations, refer to the following topics:

- *Customize gene ontology display* on page 22
- *Search Database by GO Annotation* on page 61

CHAPTER

8

WORKING WITH GENE EXPRESSION DATA

This chapter describes how to integrate Vector PathBlazer with Vector Xpression, and other expression data. Additionally, it describes overlaying gene expression data on the topology of a pathway.

Topics in this chapter include:

- *Introduction to Expression Data Import and Display* on this page
- *Interaction Between Vector PathBlazer 2.0 and Vector Xpression 3.1* on page 166
- *Creating an Template Automatically* on page 166
- *Importing Expression Data with a Template* on page 168
- *Creating a Tab-Delimited Data File of Expression Values* on page 174
- *Displaying Expression Data on Pathways* on page 178
- *Modifying Display Colors for Expression Value Ranges* on page 181

Introduction to Expression Data Import and Display

In Vector PathBlazer, gene expression data can be displayed in the context of pathway topology by linking gene names to gene products (that is, pathway components). To do this, expression data is imported into PathBlazer and links are made between genes/expression values and component names. Expression data can be forwarded to PathBlazer 2.0 directly from Vector Xpression 3.1, or intermediate tab-delimited text files can be created from other software, then imported.

Displaying expression data on pathway components in the Graphics window is a three step process.

- First, a data file that contains expression values is created (if Vector Xpression is not used)
- Second, expression data is linked to pathway components via gene names
- Finally, display colors are assigned to expression value ranges

Once the preparatory steps have been completed, expression values can be displayed on a pathway that has components in common with the genes in the expression data.

Interaction Between Vector PathBlazer 2.0 and Vector Xpression 3.1

One of the advantages of working with Invitrogen Life Science Software is that the bioinformatics software packages are designed to integrate with each other. Vector PathBlazer 2.0 includes tools for directly accessing expression data in Vector Xpression 3.1, and Vector Xpression 3.1 contains tools for exporting gene expression data directly to Vector PathBlazer 2.0.

Vector PathBlazer 2.0 is integrated with Vector Xpression 3.1 with the following features:

From Vector PathBlazer 2.0:

- You can automatically create a template that maps expression data to pathway components. The template is used to import expression data into PathBlazer. See *Creating an Template Automatically* on page 166.
- You can launch a search in Vector Xpression 3.1 for chips, expression runs and experiments containing genes coding for components of a specific pathway. See *Searching a Vector Xpression Database* on page 176.
- For a specific expression experiment in PathBlazer, you can open a corresponding object in Vector Xpression if the experiment originated in Vector Xpression. See *Opening an Experiment in Vector Xpression* on page 177.

From Vector Xpression 3.1:

- You can automatically create a template that maps expression data to pathway components. The template is used to import expression data into PathBlazer. See *Creating a Template from Vector Xpression 3.1* on page 176.
- You can send expression data directly to PathBlazer, using the template you have created. See *Sending Expression Data to PathBlazer* on page 177.
- You can launch a search in PathBlazer for components that map to expression objects in Vector Xpression. See *Finding Components in PathBlazer* on page 177.

Linking Gene Expression Data to Pathway Components

In Vector PathBlazer, you can map expression data to pathway objects in PathBlazer either automatically (recommended, where possible) or manually. Mapping the two databases is generally based on the gene names used in Vector Xpression or other expression data files and component names or component database links in PathBlazer. Links are saved as templates that can be used to import expression data files that have corresponding gene names. You can edit the mapping templates by adding additional components or deleting components. You can also share templates with other colleagues who are also using Vector PathBlazer, or import PathBlazer templates for your use.

Any pathways to which the template file applies (that is, any pathways that have components in common with the gene to component mapping) can have expression data displayed on them.

Creating an Template Automatically

To create a template automatically, associating gene names with pathway components, use the following steps.

1. In Vector PathBlazer, select **Tools >Manage Expression data > Create Expression Template.**
2. In the first screen of the Create Template Wizard, select the **PathBlazer database** to which expression data will be mapped by clicking the **Browse** button and locating the (.mdb) data-

base (*Figure 8.1*). The default location is in the C:\My Documents\My PathBlazer Data directory.

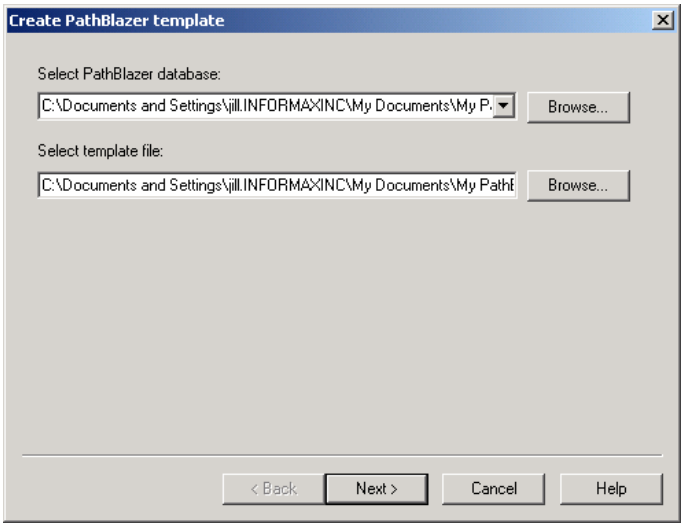


Figure 8.1 Create Template wizard, first screen for selecting the PathBlazer and expression data files to be mapped to each other

3. In the **Select Template File** field, choose the expression data file by clicking the **Browse** button and locating the (.txt) file.
4. Click **Next**.
5. In the second screen of the Wizard, the **PathBlazer database** to which components in PathBlazer are linked displays (*Figure 8.2*).

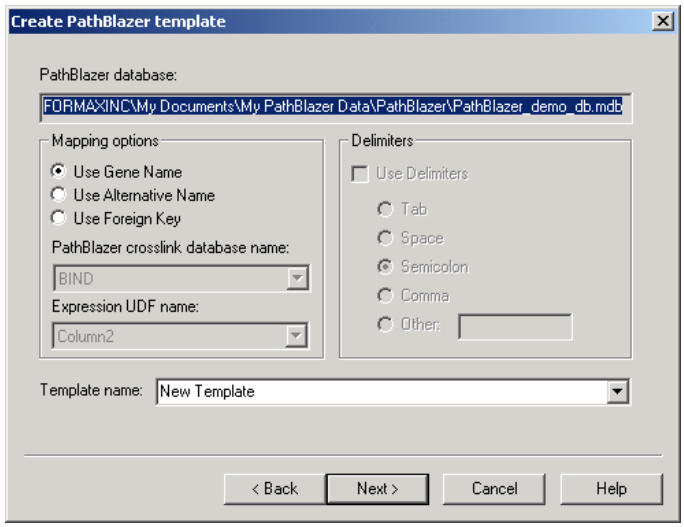


Figure 8.2 Second screen of Create Template wizard to select mapping options

In the **Mapping options** section, select from the following radio buttons:

- **Use Gene Name**--gene names are compared with the name of component. If they are the same, mapping occurs.
- **Use Alternative Name**--alternative names (synonyms) of a component are used for mapping.
- **Use Foreign Key--**if the expression file has foreign keys from external databases, such as Swiss-Prot, and the component in PathBlazer has a reference to the same object in an external database, they can be mapped. For example, if a PathBlazer component is

crosslinked to the Biomolecular Interaction Network Database (BIND) via an accession number, and that accession number is entered in a user-defined field of a gene in the Vector Xpression database, you can create a foreign key linkage in the template. If you select the **Foreign Key** option:

- Select the external database from the **PathBlazer Cross link Database Name** drop-down list. This list includes all the external databases with crosslinks to the selected PathBlazer database
- Select a column name in the Vector Xpression database from the **Expression UDF Name** drop-down list that contains the linking values to the external database. If this column contains multiple foreign keys, you can select or specify a delimiter, such as semicolon or comma, or type in a custom delimiter, such as '///'

6. In the **Template Name** text box, enter a name for the template being created. To replace or add the current information to an existing PathBlazer template, select the existing template name from the **Template Name** drop-down list.
7. Click **Next**.

The mapping is executed, and a message displays stating the number of components that were mapped. If there are conflicts, such as two or more genes being mapped to the same component, you are prompted to resolve the contradiction, i.e. select only one gene-component relationship.

Importing Expression Data with a Template

To import new expression data, use the following steps.

1. Select **Tools > Manage Expression Data > Import Expression Data**.
2. In the **Define Source** screen, select an expression data file by clicking the **Browse** button (...) and locating the appropriate file (*Figure 8.3*).
3. In the **Use Template** field, select the template you want to use with the file from the drop-down list. Click **Next**.

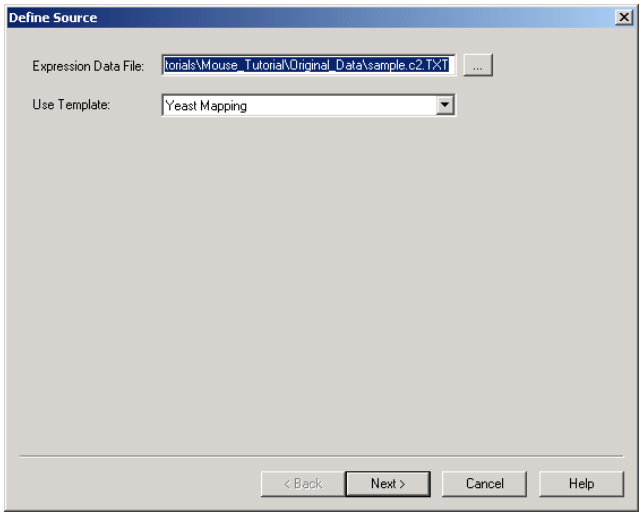


Figure 8.3 Selecting an expression template

Reminder: The template contains the component to gene name/ID mappings and the data file contains the expression values. You can apply one template to more than one data file as long as the data file contains genes that are included in the template.

4. The **Map** screen displays the current mapping contained by the template file (*Figure 8.4*). If necessary, edit the component to gene name/id mapping. If you are creating the template

automatically, you probably will have no need to perform any edits in this dialog box, however, for detailed directions about using this dialog box, see step 2. through step 5. beginning on page 172. Click **Next**.

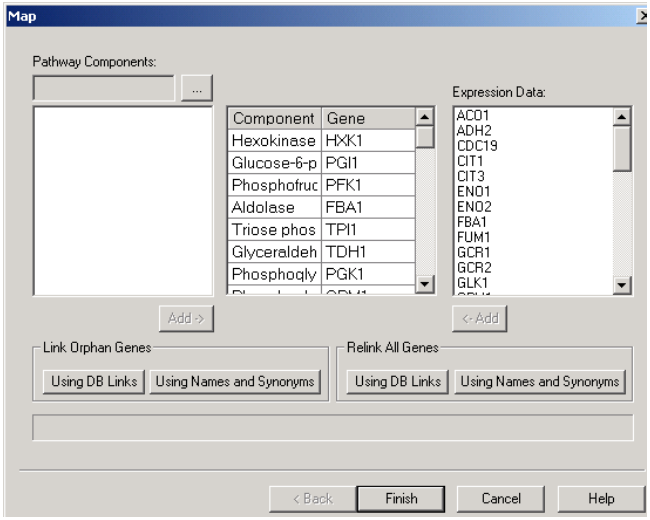


Figure 8.4 Map screen

5. In the **Specify an Import Name** screen, name the expression data file in the **Import Name** field (*Figure 8.5*). This will be the name that displays for the selected data set in the Expression Data Sets drop-down list.

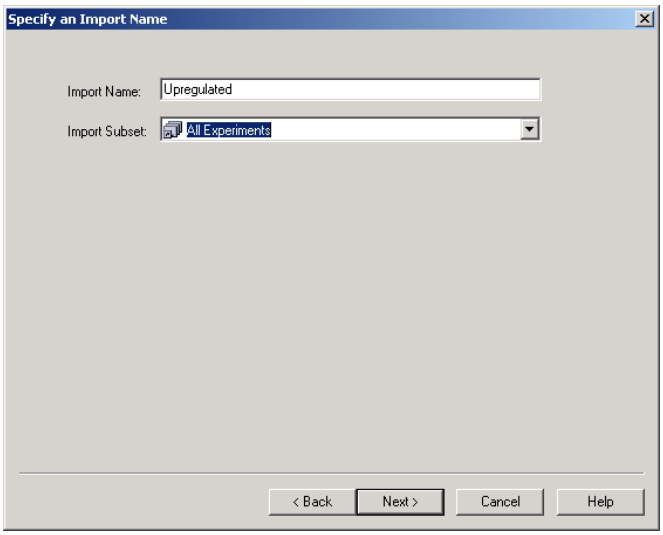


Figure 8.5 Naming the expression data set

6. In the **Import Subset** field, use the drop-down menu to select the subset into which the expression data is to be placed upon import. Click **Next**.
7. The Destination Pathways screen is for associating the expression values defined by the current expression file with one or more pathways that contain components included in the

mapping (*Figure 8.6*). Click **Add** and select a pathway from the database. Click **Finish** to save the expression values in the file and the pathway associations to the database.

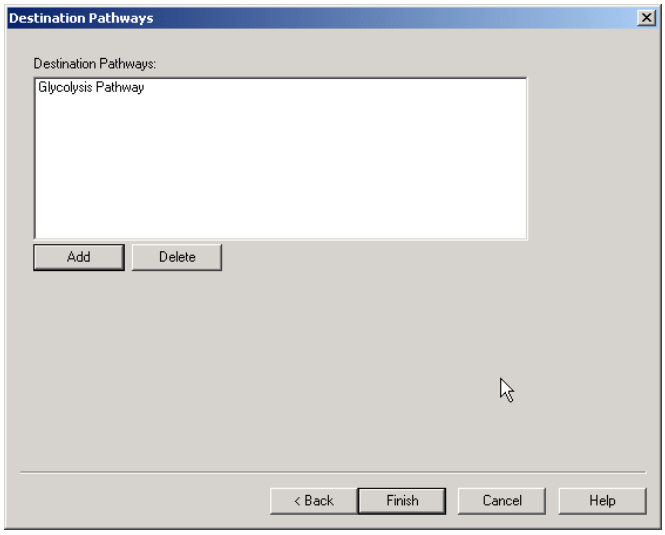


Figure 8.6 Associating pathways in the database with a data set

8. The new data set appears in the **Expression Data Sets** drop-down list in the Graphics toolbar when the pathway it is mapped to is open. It is listed permanently in the All Experiments folder in the Database Explorer.

Note: These same steps can be used to add new expression data to a pathway.

Editing a Template

Once you have established a mapping between a set of pathway components and gene names/IDs, the template you have created can be used with other expression data files that have corresponding gene names. You can also edit the mapping by adding additional components or deleting components.

To change the contents of a template, add a template, or delete a template, use the following steps.

1. Select **Tools > Manage Expression Data > Edit Expression Templates**. The **Expression Import Template Manager** opens listing any templates currently in the database (*Figure 8.7*). From this dialog box, you can add, edit, duplicate, and delete templates.

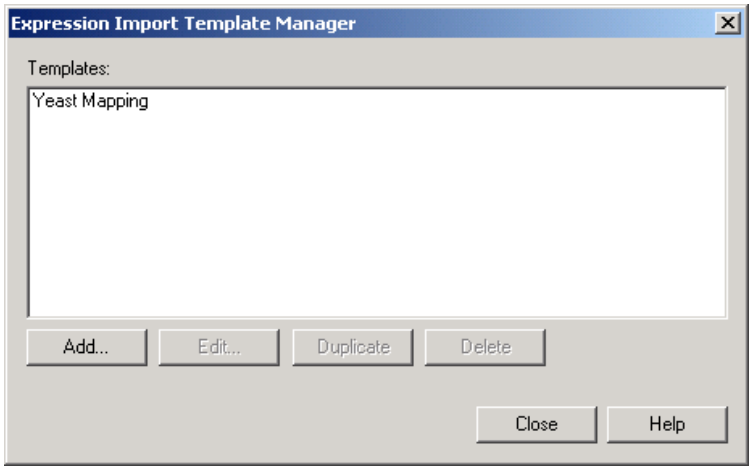


Figure 8.7 Expression Import Template Manager

2. To add more component/gene pairs to the template, select a template name from the Templates list box and click **Edit**. The Map screen opens displaying the current mapping between components and gene names/IDs (*Figure 8.8*). Use the instructions starting on step 2. on page 172 through step 5. on page 173 to modify the map.

Note: Click on the Component or Gene column headers to sort by one column or the other.

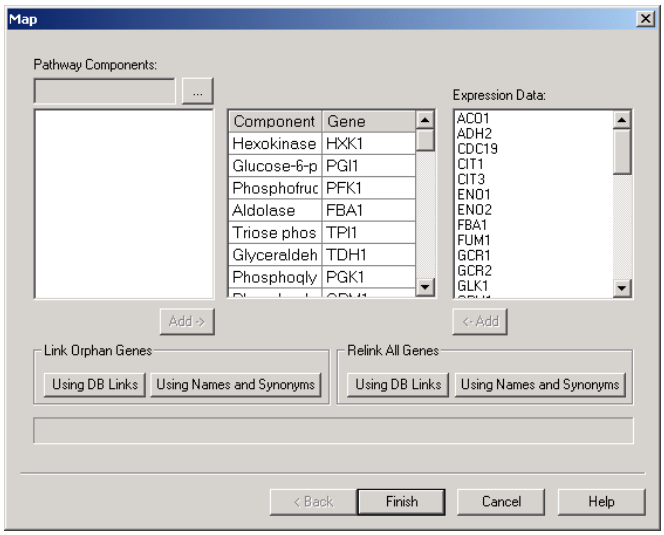


Figure 8.8 Editing a template

3. Click **Finish** to execute the edit. This returns you to the Expression Import Template Manager.
4. To add a new template, click **Add**. The Define Source screen opens where you select an expression data file. Follow the steps starting with step 2. on page 168.
5. To duplicate a template, select a template from the **Templates** list box and click **Duplicate**. Enter a new name for the template and click **OK**. The new template is added to the list box. You can then edit the mapping by clicking **Edit**.
6. To delete a template, click **Delete**. Click **OK** in the confirmation dialog box. The template is removed from the list box.
7. To close the Expression Import Template Manager dialog box, click **Close**.

Importing a Template

In Vector PathBlazer, you can import expression import templates, such as templates shared by colleagues. Import the template by selecting **Tools > Manage Gene Expression Data > Import Template**.

Mapping Database Links Manually

While creating templates automatically as described on page 166 is the ideal way to map database to each other, your file may not be compatible with that means of database object mapping. If that is not possible, you can map the database objects manually.

To associate gene names with pathway components manually, use the following steps.

1. Select **Tools > Manage Expression Data > Import Expression Data**. A wizard opens to assist you in the steps to link gene names with components. In the first screen, the tab-delimited file containing expression values is defined. Select the expression data file by clicking the **Browse** button ( ). Navigate to the file and click **Open**. The **Expression**

Data File field displays the path to the file (*Figure 8.9*). The Use Template field remains empty. Click **Next**.

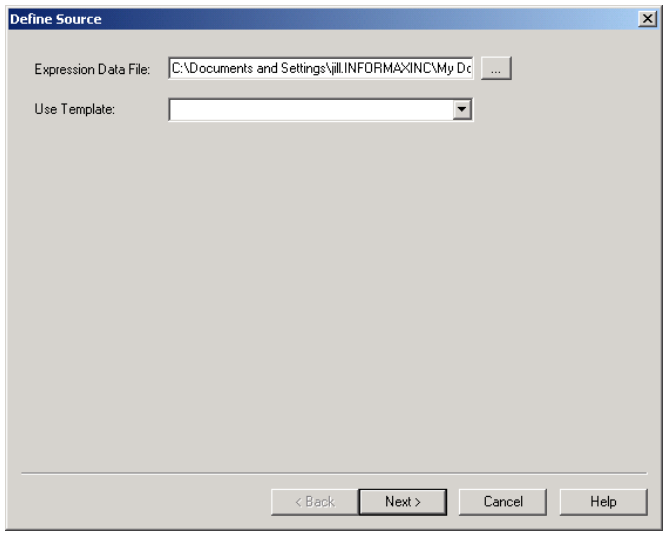


Figure 8.9 Define Source screen: Selecting an expression data file

Note: Once a mapping between a gene list and a component list is completed and saved to the database, you can use the mapping as a template and select it in the **Use Template** field. Templates are described previously in this chapter.

2. The Map screen allows a mapping to be established between pathway components on the left and gene IDs/names on the right. The gene names contained in the expression data file automatically fill in the **Expression Data** list box in alphabetical order. To select the pathway components, click the **Browse** button under **Pathway Components** and select the appropriate subset.

   In the following example, glycolysis components are organized in a subset in the database and display in alphabetical order in the Pathway Components list box (*Figure 8.10*).

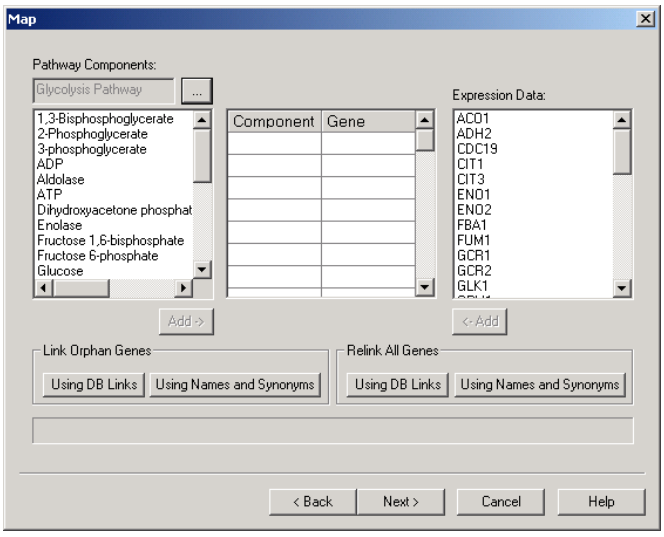


Figure 8.10 Map screen for associating components with gene names/IDs

3. Link components (that is, gene products) from the **Pathway Components** list box to gene names/IDs in the **Expression Data** list box by selecting a component and clicking the **Add** button on the left side of the screen. The component is added to the **Compound** column in the center table. In the Expression Data list box, locate the matching gene name/ID, select it, and click the **Add** button on the right side of the screen. Continue mapping components

to gene name/IDs. The following figure shows the glycolysis enzymes mapped to gene names (*Figure 8.11*). To remove a component or gene from the table, select it and press the DELETE-key.

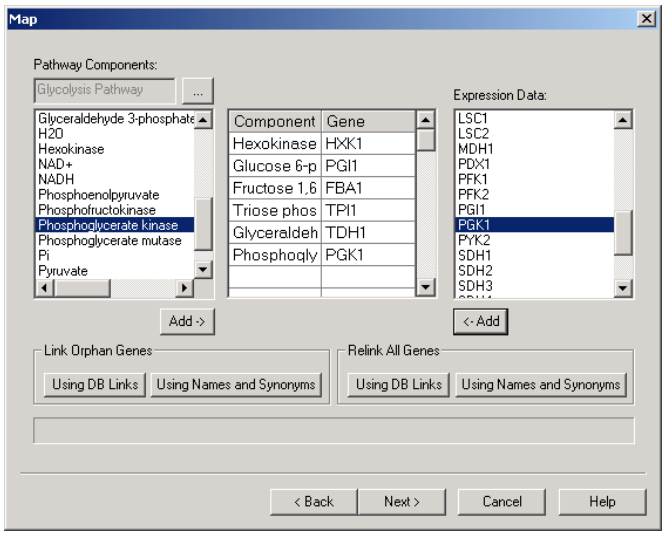


Figure 8.11 Map screen showing glycolysis enzymes mapped to gene names

If you have more than one pathway subset containing components you want to map to gene names from the expression data file currently selected, click the **Browse** button (...) uat the top of the **Pathway Components** section to select another subset and continue mapping components to the genes listed in the expression data file.

- **Link Orphan Genes**--use this button for linking genes with no known corresponding components in PathBlazer. This option leaves the rest of the template intact. Click one of the following buttons to choose the basis for the links.
  - **Using DB Links—**uses links to an external database to do the mapping. For example, a gene has a link to GB entry, A1:1234; a protein in PathBlazer has a link to the same GenBank entry. The gene and the protein will be matched in the template.
  - **Using Names and Synonyms—**uses names and synonyms to the the mapping. For example, a gene is named Top 1; a component in PathBlazer is named topoisomerase1 and has a synonym “Top 1”. They will be matched.
- **Relink All Genes**--use this button to recreate links for all genes and components as they were previously mapped. This operation creates links that are different than those that existed before.
  - **Using DB Links—**see above bullets
  - **Using Names and Synonyms—**see above bullets

4. Click **Next**.
5. The next screen allows you to specify an import name and to save the component to gene name/ID map as a template. The Import name is the name associated with the set of expression values contained in the currently selected expression data file. The template name is the name associated with the linked association table of gene names to component names. In the **Import Name** field, enter a name for the expression data values. If you want

to save the map as a template, select the **Save this map as a template** checkbox and name the template in the **Template Name** field (*Figure 8.12*). Click **Next**.

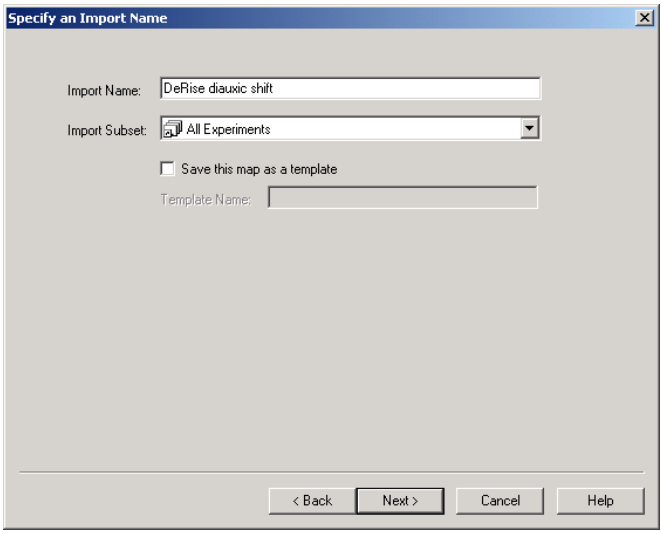


Figure 8.12 Specifying an import name for the currently expression data file

6. The next screen is for associating the expression values defined by the current file with one or more pathways that contain components included in the mapping. For example, in the previous step, the enzyme components of the glycolysis pathway were mapped to gene names and, in the following figure, the glycolysis pathway that references these components is selected. Click **Add** and select a pathway from the database. The pathway is added to the **Destination Pathways** list box. To add other pathways, click Add again. To remove a pathway association, select the pathway in the list box and click **Delete**.

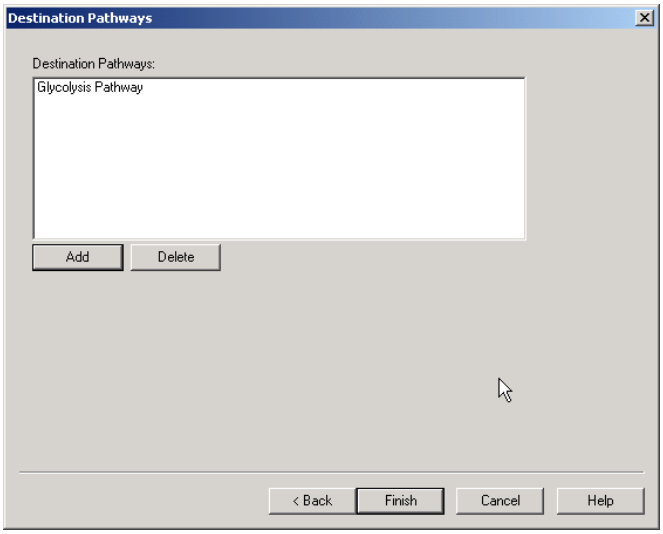


Figure 8.13 Associating expression values with pathways

To save the expression values in the data file, the component to gene name/ID map, and the pathway associations to the database, click **Finish**. Continue to the next section to assign display colors to expression value ranges.

Creating a Tab-Delimited Data File of Expression Values

If you do not have a licensed version of Vector Xpression 3.1 from which you can send expression data directly to Vector PathBlazer 2.0, you can import expression data into PathBlazer

using a tab-delimited text file. This text file contains a list of gene IDs or names and their associated expression values for each expression run included in a microarray analysis. Examples of e xpression values can be absolute values, relative values (ratio or log), P-values. The gene list and the associated expression values can be created with Vector Xpression, which can export a table containing one row for each gene to be linked to one component in the pathway (a one to one relationship). Columns in this table represent the ratios of data that will be displayed.

The expression file can contain as many columns of expression values as necessary and expression values can be from different expression experiments or Expression Runs. For example, if you want to display six time points where each value represents a normalized ratio (raw data/control) for a set of genes, you can include a column of data corresponding to each time point in the file. The format of the file that can be read by Vector PathBlazer is the following:

- The first column has the column header **Name** and each row in the column contains a gene name
- Column 2 through Column *n* contain the expression values corresponding to each Expression Run or experiment that is associated with the gene in that row. The column header can be any text string and will appear as the time point/disease state, etc. identifier in Vector PathBlazer for the values in that column.

Once a set of mappings is established, a template file can be defined from the links between a set of gene names from the gene expression file and a set of component names in Vector PathBlazer.

An example expression data file is shown in the following figure (*Figure 8.14*). The first column of the file contains a list of gene names that correspond to the enzymes in glycolysis[1]. This file is included with the Vector PathBlazer installation. It is located in a directory separate from the default database; the directory differs depending on your operating system. In Windows 2000, for example, it loads in the following directory: *C:\Documents and Settings\MyDocuments\My PathBlazer Data\DeRisi_glycolysis_TCA Expression Data.txt*. The other columns contain expression ratios for six time points (that is, six Expression Runs). The labels displaying the time points, 9 hours through 21 hours, will display as the titles of the Expression Runs in Vector PathBlazer.

	9 hours	11.5 hours	13 hours	15 hours	16.5 hours	18.5 hours	21 hours
CDC19	-0.046799	0.10046011	0.03524172	-0.2787404	-0.2936917	-0.8397514	-1.795159
PGI1	0.25493562	0.6992296	0.87479281	0.58847922	0.57035708	0.10611848	-0.0144967
GLK1	-0.4995272	-0.1105273	0.69267464	0.9716996	1.94048488	2.17512417	1.53289282
PGK1	-0.04255	0.0850837	0.50516176	0.69984198	0.5571909	0.62666702	0.14575446
GPM2	0.01022953	0.10847524	0.43968141	1.10093951	1.28455126	1.35817373	1.53552413
TPI1	-0.0605137	-0.1218064	0.12004598	0.22056738	0.12276019	0.63175696	-0.4548502
KGD2	-0.2552089	0.17974034	0.22622936	0.02555886	0.21748027	1.66687346	1.69304335
SDH4	-0.0759175	0.05301213	0.48203597	1.0740428	1.07897854	2.58559942	3.00795627
HXK1	0.13350231	0.24254444	0.2733689	1.11928892	1.6186924	3.00111532	0.8344425
HXK2	0.37024054	0.51907164	0.26144838	-0.1090048	0.21879733	0.1590258	-0.6758387
TDH3	-0.0671915	-0.0051946	0.25301778	0.3055186	0.26721618	0.56389326	0.22877938
PDX1	-0.2090392	0.06337725	0.22325653	0.18373081	0.18994701	0.4627935	0.41596839
PFK1	0.14053984	0.44347277	0.18675029	-0.0490269	-0.0448072	-0.4541662	-0.3256041
LSC2	0.16558155	0.29050538	0.67835093	1.18866527	1.69608021	2.5427947	2.36182714
ENO1	-0.0167283	0.02121283	-0.0579847	0.14575298	-0.1606347	0.55775428	0.01246041
ENO2	-0.049061	0.1431987	0.0285794	0.06889748	0.31022173	0.09780213	-0.5004593
KGD1	0.19573782	0.48567358	0.23117295	0.29444945	0.6933924	1.66465735	3.40233707
TDH1	0.02239369	0.16103742	0.53612065	0.48582798	0.5220232	0.85403061	0.01175296
TDH2	-0.0689973	-0.0156736	-0.0125684	0.078479	0.26498613	0.55451679	0.15162939
FBA1	-0.0025821	0.03332839	-0.1586238	0.0407222	0.1225418	0.30366093	-0.5819637
MDH1	0.02102616	0.09821658	0.16556883	1.09546387	1.55287337	3.18113279	2.97451711
SDH3	0.35254499	0.13607854	0.51412523	0.96216935	1.35976768	2.17717409	2.96230364

Figure 8.14 Example of expression data file that can be read by Vector PathBlazer

1. DeRisi JL, Iyer VR, Brown PO. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science. 278(5338):680-6.

Exchanging Data Between Vector PathBlazer and Vector Xpression.

Creating a Template from Vector Xpression 3.1

While directions in this chapter cover creating a expression template from PathBlazer (*Creating an Template Automatically* on page 166), you can also create a template starting from Vector Xpression 3.1. To do so, complete the following steps:

1. In the Vector Xpression Database Explorer, select **Expression Genes** from the **Tables** drop-down list, and then select the gene(s) that you want to map,

   or

   Open an Expression Runs Viewer, Runs Project Viewer or Experiment Viewer displaying data with genes that you want to map. Select the gene(s) that you want to map.

2. In the open viewer, select **Tools > Create Template in PathBlazer**.

3. This opens the Create Template Wizard. Click the **Browse** button (...) to locate and select the PathBlazer database in which you want to create the mapping template. Click **Next**.

4. The Create Template Wizard is a PathBlazer feature. At this point, continue configuring the template beginning with step 5. on page 167.

For more information about using Vector Xpression, refer to the Online Help opend from Vector Xpression 3.1 or the *Vector Xpression 3.0 User's Manual.*

Searching a Vector Xpression Database

From a specific pathway selected in PathBlazer 2.0, you can launch a search in Vector Xpression for a list of chips, Expression Runs, or Experiments containing genes coding for components of the pathway. To do so, complete the following steps:

1. From an open PathBlazer window, in the Database Explorer, select a pathway in a Pathways folder.
2. Right click on the pathway and select **Search in Vector Xpression** from the shortcut menu.
3. The Search Components dialog box that opens lists the components in the selected pathway.

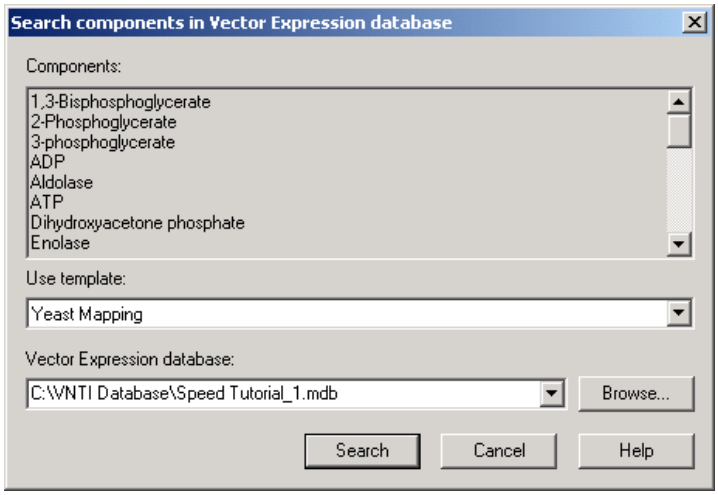


Figure 8.15 Search Components dialog box for selecting options to search a Vector Xpression database

4. In the **Use Template** field, select the template that is mapped to the expression objects linked to the listed components.
5. In the **Vector Xpression database** field, select the database to be searched.

6. Click **Search** to execute the search.

Opening an Experiment in Vector Xpression

From a specific Experiment object selected in PathBlazer 2.0, you can open the corresponding Experiment in Vector Xpression, *if the Experiment originated in Vector Xpression*. To do so, complete the following steps:

1. In the PathBlazer Database Explorer, select an Experiment in an **Experiments** folder.
2. Right click on the Experiment and select **Open in Vector Xpression** from the shortcut menu.
3. Click **Open**. Vector Xpression opens with the Experiment displayed in an Experiment Viewer.

Sending Expression Data to PathBlazer

From Vector Xpression 3.1, you can send expression data directly to Vector PathBlazer 2.0 without creating an intermediate file.

1. In the Vector Xpression Database Explorer, select **Expression Genes** from the **Tables** drop-down list, and then select the gene(s) that you want to map,

   or

   Open an Expression Runs Viewer, Runs Project Viewer or Experiment Viewer displaying data with genes that you want to map. Select the gene(s) that you want to map.
2. Select **Tools > Send Expression Data to PathBlazer**.
3. The Save Experiment(s) in PathBlazer database dialog box that opens displays the Experiment you have selected. In the **Use Template** field, select the template where the expression data is mapped.
4. In the **PathBlazer database** field, select the database where the Experiment is to be stored (*Figure 8.16*).

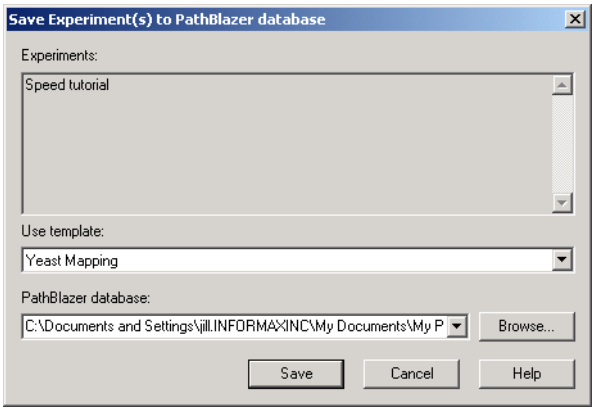


Figure 8.16 Save Experiment dialog box for selecting PathBlazer database for expression data sent from Vector Xpression

5. Click **Save**. The Experiment will now be included in the **Experiments** folders displayed in the PathBlazer Database Explorer.

Finding Components in PathBlazer

From Vector Xpression 3.1, you can launch a search in PathBlazer to find components mapped to expression data in Vector Xpression. To do so, complete the following steps:

1. In the Vector Xpression Database Explorer, select **Expression Genes** from the **Tables** drop-down list, and then select the gene(s) that you want to map,

or

Open an Expression Runs Viewer, Runs Project Viewer or Experiment Viewer displaying data with genes that you want to map. Select the gene(s) that you want to map.

2. Select **Tools > Find Components in PathBlazer**. If you have selected genes from a list, you can choose the option to search for all the listed genes or only the selected genes.
3. In the Search Genes in PathBlazer Database dialog box that opens,the selected genes are listed. In the **Use Template** drop-down menu, select the mapping template where the components are mapped.
4. In the **PathBlazer Database** drop-down menu, select the PathBlazer database to be searched.
5. Click **Search**. This opens the PathBlazer application and the search is transferred to the PathBlazer search engine. It includes prompts for further defining the scope of the search. For more information, see *Searching Objects in the Database and Creating Subsets* on page 54.

Search results display in the same format as do object searches launched from PathBlazer. For more information, see *Search Results* on page 58

For more information about working in Vector Xpression, refer to the *Vector Xpression 3.0 User's Manual* and the *Vector Expression 3.1 User's Manual Addendum*.

Displaying Expression Data on Pathways

Once components have been mapped to gene names/IDs and colors have been assigned to expression value ranges, you can display expression values on pathway components. To do so, you must associate expression experiments with pathways. Use one of the following methods to initiate the display:

- Select an Experiment subset in the Database Explorer List Pane. Right click on the Experiment and select **Associate With**. In the dialog box that opens, locate and check one or more pathways you want to associate with the Experiment. Click **Select**. The association are saved to the database.
- Open a pathway. Click on an Experiment in the Database Explorer List Pane and drag it onto the pathway in the Graphics Window.
- Select an expression data set from the **Expression Data Set** drop-down list in the Graphics toolbar (*Figure 8.17*). (If no data sets are associated with a pathway, **None** is the only option in the drop-down list.) Once a data set has been selected, select an expression run from the **Expression Runs** drop-down list. (Those displayed are associated with the selected Expression Data Set.) Expression Runs are listed according to the title of the column headers in the data file. If no headers are present, the expression runs are labeled generically as Run 1, Run 2, etc.

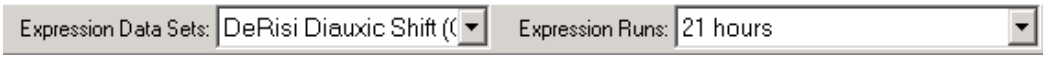


Figure 8.17 Drop-down lists in the Graphics toolbar for displaying expression data. The DeRisi data is use as an example dataset.[1]

If there are pathway components that map to genes in the Experiment, color-coded rectangles representing the expression values of the genes display on the pathway components (Discovery View) or the pathway reaction nodes (Metabolic View). The color(s) in the square correspond to the expression value range displayed in the Expression Palette. The associated Experiment(s) also display on the Expression Data tab in the Pathway Properties dialog box.

1. DeRisi JL, Iyer VR, Brown PO. 1997. *Exploring the metabolic and genetic control of gene expression on a genomic scale*. Science. 278(5338):680-6.

Default Display Colors for Expression Values

To view expression data on pathway components, display colors are assigned to expression value ranges. These settings are independent of a particular expression data file or pathway and are used to display all expression data on components (in **Discovery View**) or reaction nodes (in **Metabolic View**) regardless of source data files or associated pathways. For example, you might be measuring expression changes in a normal versus a disease state. Expression values for the normal state may be between 0 and 0.5 while the disease state results in a marked upregulation of all genes to a range of 1 to 1.5. You associate the color blue with the 0 to 0.5 range and red with the 1 to 1.5 range. When you view the expression data on the pathway components, components associated with expression genes are colored blue when expression values for the normal state are displayed and red when expression values for the disease state are displayed.

To view the color key of the range or expression values in the associated experiment, open the Expression Palette by selecting **View > Expression Palette** (*Figure 8.18*). The Expression Palette is anchored on the right side of the screen by default but can be converted to an independent window by clicking on the double-line on the top of the window and dragging and dropping it anywhere on the screen when its borders retract to a smaller rectangle. To re-anchor it on the right side of the screen again, drag it to the right and drop it when its borders expand to fill the right side or double-click on its title bar to return it to the right side.

If the palette is turned on, it is printed and/or exported in the image file.

A small graphic display in the format of a bar diagram with colors coded from the expression colors palette can display as a label next to a component in the Graphics Pane. These small graphs are designed to give you a quick grasp of expression differences (*Figure 8.18*). *The Y axis of these graphs are expressed in "relative units" rather than actual expression values.* To show/hide these graphs, select **View > Show/Hide All Expression Difference Labels**.

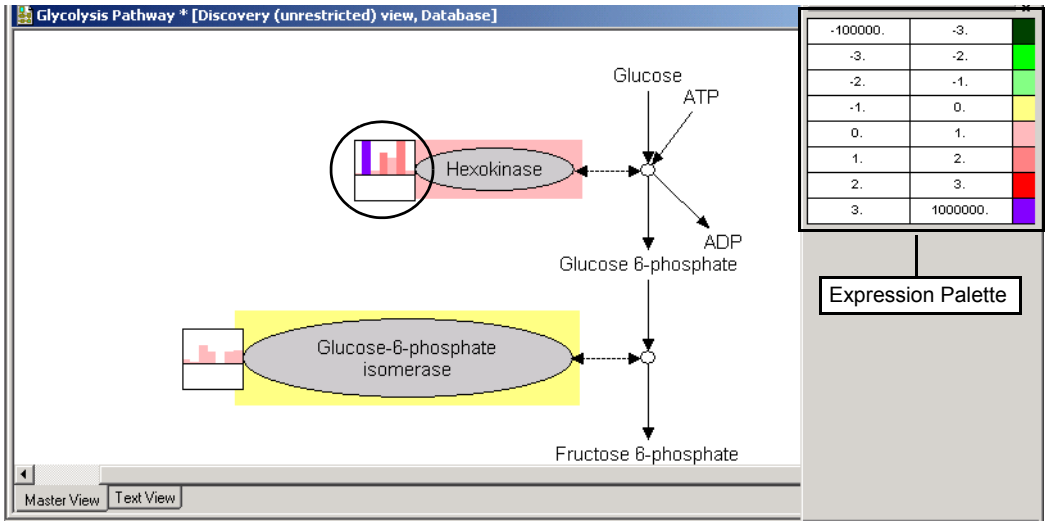


Figure 8.18 Small bar graphs representing expression values (circled) can display next to their corresponding components in the Graphics window. The Expression Palette displays to the upper right of the Graphics window.

Displayed expression graphs are printed with the Graphics window and/or copied and exported with image files.

The following figure shows glycolysis displayed in Discovery View (*Figure 8.19*). In Discovery View, the colors assigned to the expression values ranges are associated with the actual component (the enzymes in this example). In Metabolic View, the enzymes in the pathway display with the reaction nodes and the expression values display on the reaction nodes.

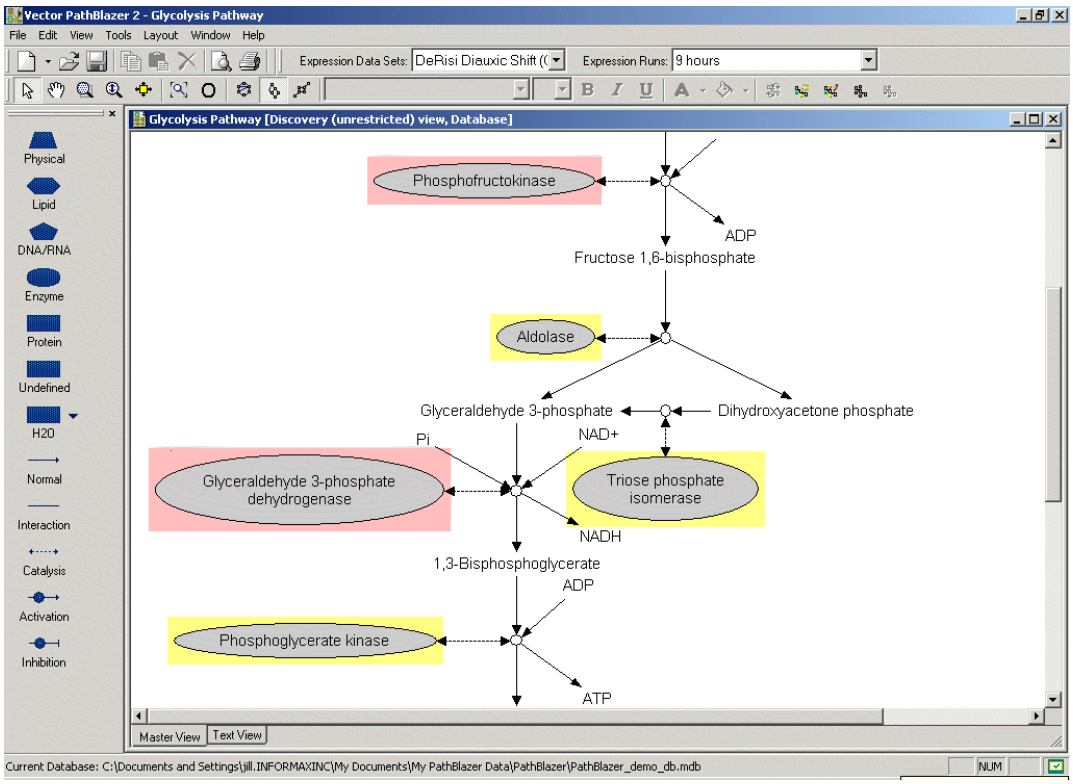


Figure 8.19 Expression values display on enzyme components for Glycolysis in Discovery View

6. Pause the cursor over a component to display a tool tip that contains the component name, the expression run name, the gene name/ID, and the expression value (*Figure 8.20*).

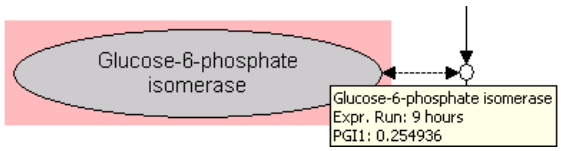


Figure 8.20 Tool tip that display expression information about a component

7. Select a different expression run from the Expression Runs drop-down list on the Graphics toolbar to display another set of expression values on the pathway (*Figure 8.21*).

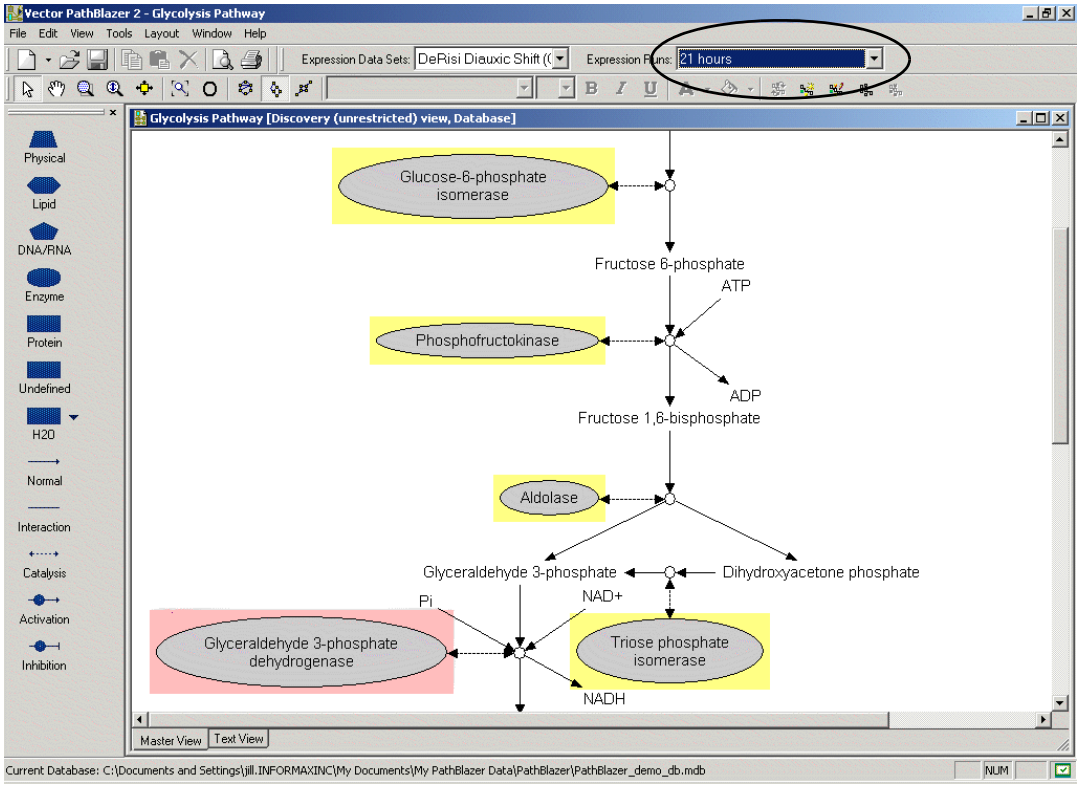


Figure 8.21 Different expression runs can be displayed on pathway components

Modifying Display Colors for Expression Value Ranges

Use the following steps to modify colors for expression value ranges.

1. Select **Tools > Options** and select the **Set Expression Data Ranges** tab. The tab contains the columns **Start**, **End**, and **Color** and initially contains the default values shown in *Figure 8.22*.

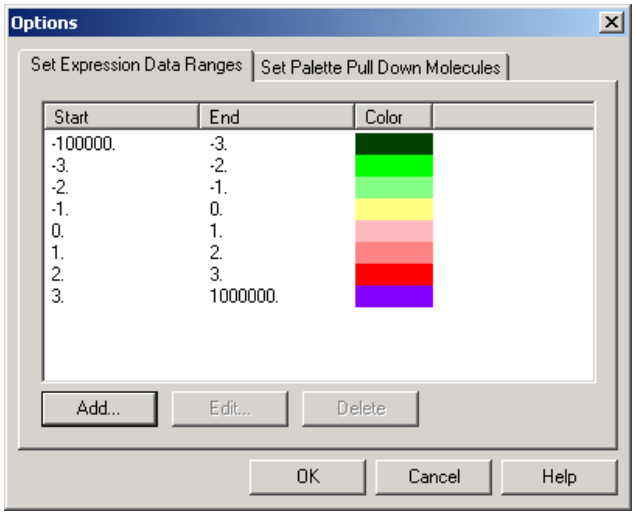


Figure 8.22 Expression Data Ranges tab for associating expression value ranges with display colors

2. To define a new range, click **Add** to open the **Expression Data Range** dialog box (*Figure 8.23*). Define the start and end value of the range in the **Start Value** and **End Value** fields.

The Start Value is defined as greater than or equal to (>=) and the End Value is defined as less than (<). Assign a color to this range by clicking the **Browse** button in the **Color** field. Select a color from the palette and click **OK**. Click **OK** in the Expression Data Range dialog box.

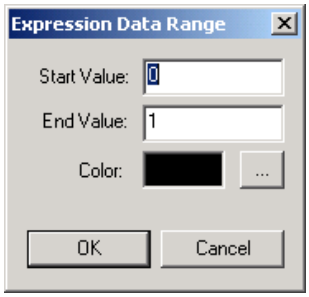


Figure 8.23 Dialog for defining range values and colors

3. Add additional ranges by repeating the instructions in step 2. Edit a color or range by selecting the definition, clicking **Edit**, and making the change in the Expression Data Range dialog box. Delete a definition by selecting it and clicking **Delete**.
4. Continue to the next section to display expression data on pathway components by the associated color and range.

APPENDIX

A

LICENSE MANAGER

Once you have installed Vector PathBlazer, you will need to license the application to be able to use it.

To satisfy the needs of users in different industrial, scientific or educational environments, Invitrogen has designed four types of Vector PathBlazer licenses. These are all administered through the License Manager.

- **Static License**: Purchased by one user for installation on one computer
- **Dynamic License (DLS)**: A license that is installed on a server and issued by that server to client Vector Advance computers. DLS licenses are shared by a specified number of users or “seats,” with the number of users at any one time being limited to the number of “licenses” specified in the contract.
- **Trial License**: Allotted to a potential purchaser of Vector software for a specified number of days, during which the user can review and use the software within certain limits.
- **Demo Mode**: For the purposes of demonstrating the Vector software. Some functionality is disabled in Demo Mode.

When you open the Vector PathBlazer software, a checkmark icon, such as this ( ), at the bottom right corner on the Status Bar shows the current license status. Pause the cursor arrow over the button, and a pop-up label displays the license status.

- Green checkmark = active Static License
- Green, *blinking* checkmark = active Trial or Dynamic License
- Red, *blinking* X = the application is not licensed; running in Demo mode.

License Manager does not open automatically when you install Vector PathBlazer (or Vector Advance) on your computer. You must open License Manager manually. To open License Manager, select it from the **Start** menu: **Start > Programs > InforMax 2003 > Vector PathBlazer 2 > License Manager** or click **Help > License** in the PathBlazer Viewer once you have opened it.

License Manager Dialog Box

The License Manager has three tabs, the **Contact Us** tab, the **Personal** tab, and the **Applications** tab.

Contact Us Tab

The License Manager opens by default to the **Contact Us** tab (*Figure A.1*). This tab summarizes your Vector software licensing agreement. Additionally, it provides information for upgrading your Vector application license and contacting Invitrogen.

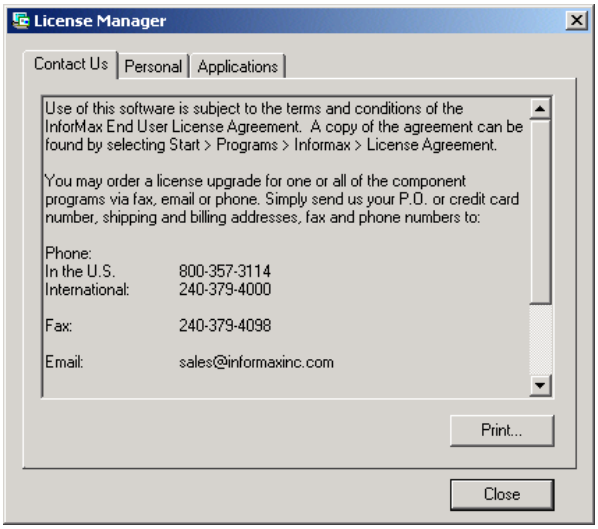

License Manager
Contact Us
Personal
Applications
Use of this software is subject to the terms and conditions of the InforMax End User License Agreement. A copy of the agreement can be found by selecting Start > Programs > Informax > License Agreement.
You may order a license upgrade for one or all of the component programs via fax, email or phone. Simply send us your P.O. or credit card number, shipping and billing addresses, fax and phone numbers to:
Phone:
In the U.S. 800-357-3114
International: 240-379-4000
Fax: 240-379-4098
Email: sales@informaxinc.com
Print...
Close

Figure A.1 License Manager (Contact Us tab)

Personal Tab

The **Personal** tab (*Figure A.2*) provides text boxes for entering personal information. Once entered on this tab, when you click your license choice on the **Applications** tab, your entries are automatically entered on the license application.

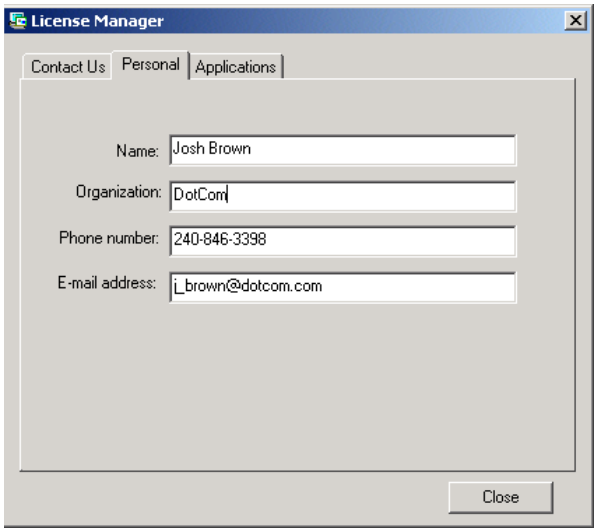


Figure A.2 License Manager (Personal tab)

Applications Tab

The **Applications** tab (*Figure A.3*) indicates the type of License currently in effect for each Vector NTI Advance application, as well as for Vector PathBlazer and Vector Xpression.

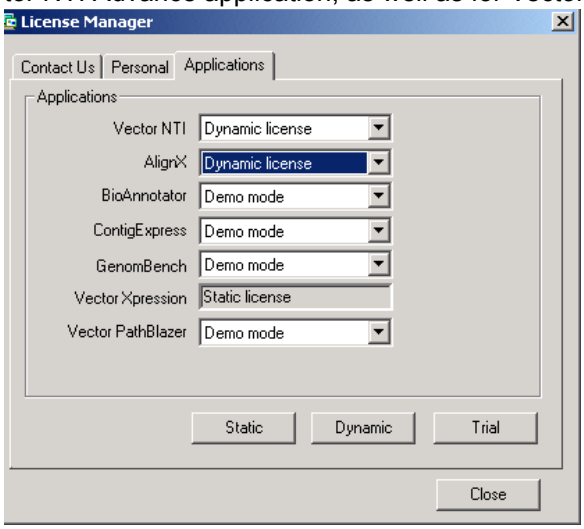


Figure A.3 License Manager (Applications tab)

For a new installation or update of a previously unlicensed installation, License Manager opens in Demo mode for all applications.

For Dynamic and Trial licenses, if you are not licensing the entire software package using the same type of license, on the **Applications** tab click in the license-type text box of the application for which you wish to specify a license. Click the down-arrow to extend the drop-down menu and select the appropriate license type(s).

Click the button appropriate for the license type you want to register. Each option is described in the following sections.

Static License Dialog Box

To configure your static license, click the **Static** button at the bottom of the **Applications** tab (*Figure A.3*). This opens the Static License dialog box (*Figure A.4*)..

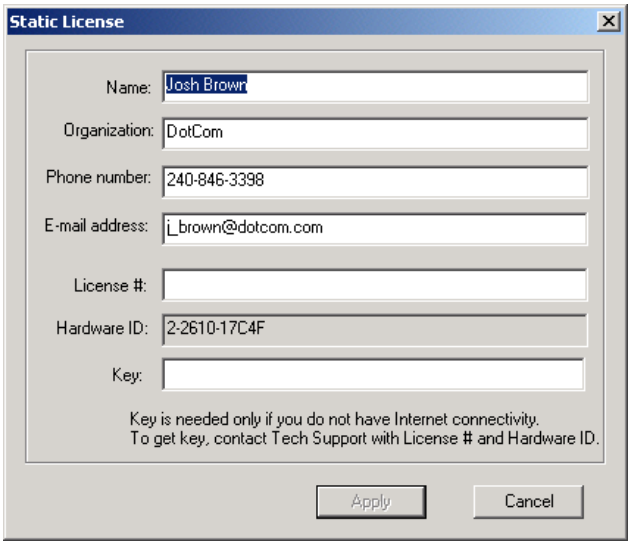


Figure A.4 Static License dialog box

Enter your name, organization, phone number and email address in the appropriate fields. This sets the user information in Vector PathBlazer.

Note: If you already entered your personal information on the **Personal** tab, it should appear here when you open this dialog box.

In the **License #** field, enter your Vector PathBlazer static license number provided in the letter accompanying your CD ROM and/or manual. Click the **Apply** button. Your software is registered immediately.

If the registration fails because of a missing connection to the Invitrogen licensing server, an appropriate message immediately displays. In such a case, you can contact Invitrogen/InforMax Technical Support or Sales (see and provide them your computer's hardware ID and your license number. They

Once you receive the registration key, enter the key in the **Key** text box of the Static License dialog box. Make sure the License Number is entered appropriately, and click Apply. If the Key matches your license number and computer hardware ID, the license is registered. No connection to the Internet is required in this case.

Notes:

- Once you have applied your static license, notice that the **Applications** tab reflects your static license status.
- If you want to reset your static license, type ***Unregister*** in the License Number field and click Apply. You will be warned that you are trying to reset your static license and asked if you want to continue. If you answer **Yes**, the application will reset your license and will send proof of this operation to the Invitrogen server. If the connection to the server fails, you will receive notice of this.

Dynamic License Dialog Box

To configure your Dynamic license, click the **Dynamic** button at the bottom of the **Applications** tab (*Figure A.3*). This opens the Dynamic License dialog box (*Figure A.5*):

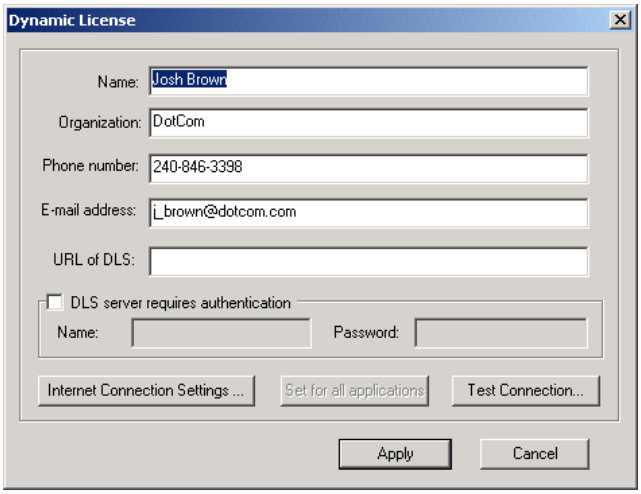


Figure A.5 Dynamic License dialog box

Enter your name, organization, phone number and email address in the appropriate fields. This sets the user information in Vector PathBlazer.

Note: If you already entered your personal information on the **Personal** tab, it should appear here when you open this dialog box.

In the **URL of DLS** text box, enter the DLS server URL supplied by the DLS administrator at your site. If your DLS server requires a password, make sure the authentication settings are filled in appropriately.

Press the **Internet Connection Settings** button to configure your connection settings and to enter server proxy information, if a firewall is used at your site. See the Internet Connection Settings section on page 188 for more information.

For information on the **Test Connection** button, see page 188.

Once you have configured the **Dynamic License** dialog box parameters, to set all Vector applications to Dynamic License, press the **Set For All Applications** button. Once you do this, when you close this dialog box, the Applications tab now shows Dynamic License for all applications.

Note: When you set Dynamic licenses for all applications, this operation only applies for those applications for which you do not have a Static License.

Press the **Apply** button to execute the dynamic license configuration.

Trial License Dialog Box

To configure a trial license, click the **Trial** button at the bottom of the **Applications** tab of License Manager (*Figure A.3*). This opens the Trial License dialog box (*Figure A.6*):

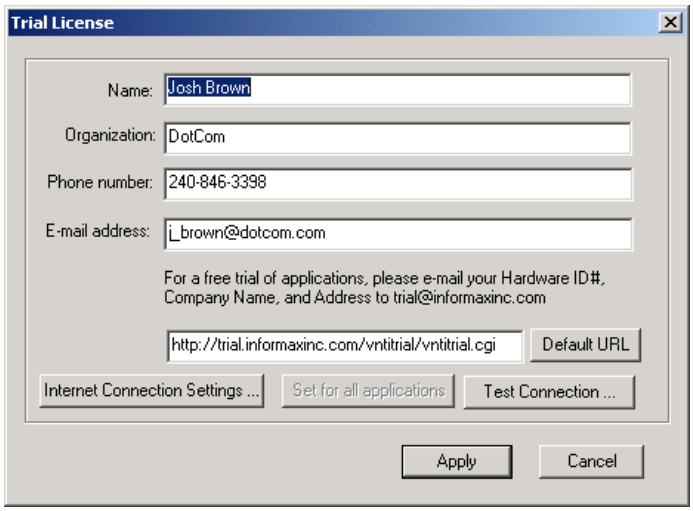


Figure A.6 Trial License dialog box in License Manager

Enter your name, the name of your organization, phone number, and email address in the appropriate fields.

Note: If you already entered your personal information on the **Personal** tab, it should appear here when you open this dialog box.

Enter the server URL or click the **Default URL** button to enter it automatically.

Press the **Internet Connection Settings** button to configure your connection settings and to enter server proxy information, if a firewall is used at your site. See the Internet Connection Settings section on page 188 for more information.

For information on the **Test Connection** button, see page 188.

Important: Trial licenses are served from Invitrogen. To receive your trial license, send the Hardware ID from the **Trial License** dialog box to trial@informaxinc.com with your personal information. You will generally receive a prompt reply, usually within one business day. Once you have received the reply, testing the connection (see following section) will show that licenses are available.

Once you have tested the connection and have a Trial License available, the **Set for All Applications** button becomes available. Click this button to set all Vector applications applications to Trial Licenses. Once you do this, when you close this dialog box, the Applications tab now shows Trial License for all applications.

Note: When you set Trial Licenses for all applications, this operation only applies for those applications for which you do not have a Static License.

Testing the License Server Connection (Dynamic and Trial Licenses)

In both the Dynamic License and Trial License dialog boxes, press the **Test Connection** button to review the status of your connection. This opens the Server Connection Tester dialog box (*Figure A.7*).

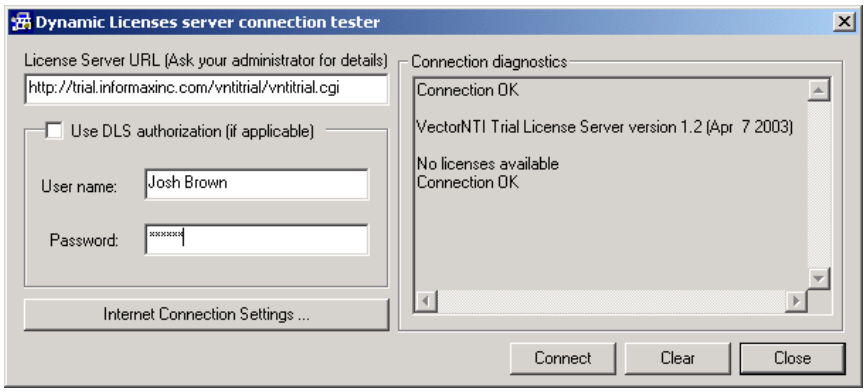


Figure A.7 Dynamic License Server Connection Tester dialog box

The status of the connection displays in the right-hand panel. For a trial license, it will report that there are no licenses available until you request a trial license (see *Trial License Dialog Box* above). If the server requires a password, it must be entered into the corresponding text box in this dialog box. If you want to alter your proxy settings, press the **Internet Connection Settings** button (see next section). Once the settings are reconfigured, press the **Connect** button to test the connection using the new settings.

Internet Connection Settings (Dynamic and Trial Licenses)

For Dynamic or Trial licenses, press the **Internet Connection Settings** button in the Dynamic License Server Connection dialog box. This opens the Internet Settings dialog box where you can alter your proxy settings (*Figure A.8*):

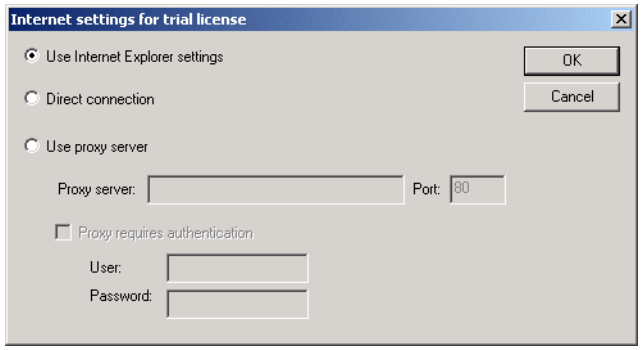


Figure A.8 Internet Settings dialog box

The Internet Settings dialog box allows you to set your connection parameters. If the **Use Internet Explorer settings** button is selected, License Manager will attempt to make the connection using your default settings. If default detection is not successful, you can either choose the

Direct connection button if you do not have a proxy or choose the **Use proxy server** button and specify the proxy name, port and password information.

Press the **OK** button to return to the Dynamic License Server Connection Tester dialog box.

APPENDIX

B

DTD FOR DATA IMPORT

This appendix includes the Document Type Definition (DTD) for mapping proprietary data to a PathBlazer-formatted XML file for import.

```
<!ELEMENT storage (list_of_substances, list_of_interactions)>
<!ATTLIST storage ID ID #REQUIRED>

<!ELEMENT list_of_substances (substance+)>

<! -- ==== Description of Substance ========================== -->
<!ELEMENT substance (list_of_origin_accesses?,
creator?,
create_date?,
update_date?,
list_of_hyperlinks?,
        synonyms?,
        type,
group_name?,
list_of_subcomponents?,
definition_of_locations?,
list_of_pathways_names?,
list_of_annotations?,
list_of_reference_accesses?,
```

```
comments?,
            list_of_formulas?)>
<!ATTLIST substance ID ID #REQUIRED>

<! -- Description of OriginAccess ------------------ -->
<!ELEMENT list_of_origin_accesses (origin_access*)>
<!ELEMENT origin_access (type_of_data?,database,access,item_URL?,extra_data?)>
<!ELEMENT type_of_data (#PCDATA)>
<!ELEMENT database (#PCDATA)>
<!ELEMENT access (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT extra_data (#PCDATA)>

<!ELEMENT synonyms (name*)>

<! -- NMTOKENS's string is represented like "class|subclass|..." -->
<!ELEMENT type NMTOKENS >

<! -- distributed ontology table -->
<!ELEMENT group_name (#PCDATA)>
<!ELEMENT list_of_subcomponents (name*)>

<!ELEMENT list_of_locations (location*)>
<! -- Description of Location ---------------------- -->
<!ELEMENT location (species, tissue, celltype, cell_compartment, stage)>
<!ELEMENT species NMTOKEN >
<!ATTLIST species Op CDATA #REQUIRED>
<!ELEMENT tissue NMTOKEN >
<!ATTLIST tissue Op CDATA #REQUIRED>
<!ELEMENT celltype NMTOKEN >
<!ATTLIST celltype Op CDATA #REQUIRED>
<! - Description of CellCompartment ----------- -->
<!ELEMENT cell_compartment (item, parts, parts_location)>
<!ELEMENT item NMTOKEN >
<!ATTLIST item Op CDATA #REQUIRED>
<!ELEMENT parts NMTOKEN >
<!ATTLIST parts Op CDATA #REQUIRED>
```

```
<!ELEMENT parts_location NMTOKEN >
<!ATTLIST parts_location Op CDATA #REQUIRED>
<! --- ---------------------------------------- -->
<!ELEMENT stage NMTOKEN >
<!ATTLIST stage Op CDATA #REQUIRED>
<! -- ---------------------------------------------- -->

<!ELEMENT list_of_pathways_names (pathway_name*)>
<!ELEMENT pathway_name (#PCDATA)>

<!ELEMENT list_of_annotations (annotation*)>
<!ELEMENT annotation (#PCDATA)>

<!ELEMENT list_of_reference_accesses (db_reference*)>
<! -- see origin_access -->
<!ELEMENT db_reference (type_of_data?,database,access,item_URL?,extra_data?)>

<!ELEMENT comments (#PCDATA)>

<!ELEMENT list_of_formulas (formula*)>
<!-- Description of formula -->
<!ELEMENT formula (SMILE?)>
<!ATTLIST formula expr CDATA #REQUIRED>
<!ELEMENT SMILE (#PCDATA)>

<! -- ========================================================= -->

<!ELEMENT list_of_interactions (interaction | reaction | pathway)*>

<! -- ==== Description of Interaction ========================= -->
<!ELEMENT interaction (list_of_origin_accesses?,
creator?,
create_date?,
update_date?,
list_of_hyperlinks?,
                synonyms?,
                type,
```

```
group_name?,
list_of_subcomponents?,
definition_of_locations?,
list_of_pathways_names?,
list_of_annotations?,
list_of_reference_accesses?,
comments?,
list_of_conditions?,
list_of_diseases?,
reversible,
effect?,
confidence_level,
BioNet)>
<!ATTLIST interaction ID ID #REQUIRED>

<! -- list_of_conditions ---------------------------------- -->
<!ELEMENT list_of_conditions (condition*)>
<!ELEMENT condition CDATA #REQUIRED>
<!ATTLIST condition type CDATA #REQUIRED>

<! -- list_of_diseases ---------------------------------- -->
<!ELEMENT list_of_diseases (disease*)>
<!ELEMENT disease (database?,access?,item_URL?)>
<!ATTLIST disease name CDATA #REQUIRED>


<!ELEMENT reversible ("Yes"|"No")>

<!ELEMENT effect (#PCDATA)>

<!ELEMENT confidence_level (#PCDATA)>

<! -- ==== Description of Reaction ============================== -->
<!ELEMENT reaction (list_of_origin_accesses?,
creator?,
create_date?,
update_date?,
```

```
list_of_hyperlinks?,
                synonyms?,
                type,
group_name?,
list_of_subcomponents?,
definition_of_locations?,
list_of_pathways_names?,
list_of_annotations?,
list_of_reference_accesses?,
comments?,
list_of_conditions?,
list_of_diseases?,
reversible,
effect?,
confidence_level,
BioNet,
                list_of_formulas?,
list_of_constants?)>
<!ATTLIST reaction ID ID #REQUIRED>

<! -- list_of_constants ---------------------------------- -->
<!ELEMENT list_of_constants (constant*)>
<!ELEMENT constant CDATA #REQUIRED>
<!ATTLIST constant type CDATA #REQUIRED>

<! -- ==== Description of Pathway ============================== -->
<!ELEMENT pathway (list_of_origin_accesses?,
creator?,
create_date?,
update_date?,
list_of_hyperlinks?,
                synonyms?,
                type,
group_name?,
list_of_subcomponents?,
definition_of_locations?,
list_of_pathways_names?,
```

```
list_of_annotations?,
list_of_reference_accesses?,
comments?,
list_of_conditions?,
list_of_diseases?,
reversible,
effect?,
confidence_level,
BioNet,
validity?)>
<!ATTLIST pathway ID ID #REQUIRED>

<!ELEMENT validity (universally accepted|novel|hypothetical|doubtful|experimental-test-
dummy)>

<! -- ==== BioNet structure ================================== -->
<!ELEMENT BioNet (list_of_agents, list_of_actions, list_of_arcs)>
<!ATTLIST BioNet ID ID #REQUIRED>

<! -- Description of Agent ---------------------- -->
<!ELEMENT list_of_agents (agent*)>
<! -- ID: should be unique into current 'BioNet' only -->
<!ELEMENT agent (role, substance_ref)>
<!ATTLIST agent ID ID #REQUIRED>

<!ELEMENT role (educt|product|catalyst|inhibitor|intermediate|none)>

<! -- IDREF: reference to substance placed into 'list_of_substances'-->
<!ELEMENT substance_ref "substance">
<!ATTLIST substance_ref ref IDREF #REQUIRED>
<! -- ------------------------------------------- -->

<! -- Description of Action --------------------- -->
<!ELEMENT list_of_actions (action*)>
<! -- ID: should be unique into current 'BioNet' only -->
<!ELEMENT action (interaction_ref)>
<!ATTLIST action ID ID #REQUIRED>
```

```
<! -- IDREF: reference to any kind of interactions -->
<! -- placed into 'list_of_interactions' -->
<!ELEMENT interaction_ref (interaction|reaction|pathway) "reaction">
<!ATTLIST interaction_ref ref IDREF #REQUIRED>
<! -- ------------------------------------------- -->

<! -- Description of Arc ------------------------ -->
<!ELEMENT list_of_arcs (conf_arc*)>
<! -- IDREF: references to agents/actions,                    -->
<! -- placed into 'list_of_agents'/'list_of_actions' respectively -->
<!ELEMENT conf_arc (bidirect,
  type,
  weight,
  conf_level,
  expression?)>
<!ATTLIST conf_arc from IDREF #REQUIRED
 to   IDREF #REQUIRED>

<!ELEMENT bidirect (Yes|No)>
<!ELEMENT type (ordinary|enabling|disabling)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT conf_level (#PCDATA)>
<! -- expression should be conformable to Perl grammar -->
<!ELEMENT expression (#PCDATA)>
<! -- ------------------------------------------- -->

<! -- ========================================================= -->
```

APPENDIX

C

REFERENCES

This appendix contains a list of references to locations and citations where you can obtain more information about key concepts in Vector PathBlazer.

General

Fell DA. *Understanding the Control of Metabolism*. Portland Press, 1996.

Girault C and Valk R. *Petri Nets for Systems Engineering*. Springer Verlag, 2002. First Edition.

Kanehisa M. *Post-genome Informatics*. Oxford University Press, 2000.

Kitano H. *Foundations of Systems Biology*. The MIT Press, 2001.

Peterson JL. *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs, N.J.: Prentice-Hall, 1981.

von Bertalanffy L. *General System Theory*. Brazilier, New York, 1968.

KEGG

Description

KEGG (Kyoto Encyclopedia of Genes and Genomes) is an effort to computerize current knowledge of molecular and cellular biology in terms of the information pathways that consist of interacting molecules or genes and to provide links from the gene catalogs produced by genome sequencing projects.

URL

http://fire2.scl.genome.ad.jp/kegg/

References

Goto S, Okuno Y, Hattori M, Nishioka T, and Kanehisa M. 2002. LIGAND: Database of Chemical Compounds and Reactions in Biological Pathways. Nucleic Acids Research 30(1):402-4.

Kanehisa M and Goto S. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Research, 28:29-34.

Licensing Information

Academic users may freely download the KEGG data as provided at the GenomeNet ftp site at ftp://ftp.genome.ad.jp/pub/kegg/.

Non-academic users may also download the KEGG data from this ftp site as long as they are used for internal research purposes.

For more information, see http://fire2.scl.genome.ad.jp/kegg/kegg5.html.

BIND

Description

The Biomolecular Interaction Network Database (BIND) is a database designed to store full descriptions of interactions, molecular complexes, and pathways.

URL

http://www.bind.ca/index.phtml

References

Bader GD, Donaldson I, Wolting C, Ouellette BF, Pawson T, and Hogue CW. 2001. BIND--The Biomolecular Interaction Network Database. Nucleic Acids Research, 29(1):242-45.

Licensing Information

There are no license conditions attached to the use of the BIND database. All data records in the public BIND database are in the public domain.

BioCyc

Description

A collection of Pathway/Genome Databases make up the BioCyc Knowledge Library. The genome and metabolic pathways of a unique organism are represented in each database in the BioCyc collection. The MetaCyc database, however, is an exception in that it is a reference source on metabolic pathways from many organisms.

The above text is paraphrased from the BioCyc website, listed below.

URL

http://www.biocyc.org

Reference

Karp, P.D., Riley, M., Saier, M., Paulsen, I.T., Collado-Vides J., Paley, S.M., Pellegrini-Toole, A., Bonavides C., Gama-Castro S. The Ecocyc database, Nucleic Acids Research, 30(1):56 2002.

Licensing Information

http://www.biocyc.org

Transpath

Description

The TRANSPATH® Professional database is a repository of data for molecules participating in signal transduction and the reactions they undergo, thus spanning a complex network of interconnected signalling components. TRANSPATH® Professional focuses on signalling cascades that aim at transcription factors and thus alter the gene expression profile of a given cell.

TRANSPATH® Professional is the resource of choice in disclosing the upstream regulators and downstream targets of each molecule in the regulatory network. Connected and integrated with the TRANSFAC® Professional database, TRANSPATH® Professional bridges the gap between extra cellular signal molecules (such as hormones, cytokines etc.) and the genes responding to these triggers.

The above text is taken from the TransPath website, listed below.

URL

http://transpath.gbf.de

Reference

Schacherer, F., Choi, C., Gotze, U., Krull, M., Pistor, S., Wingender, E. The TRANSPATH signal transduction database: a knowledge base on signal transduction networks. Bioinformatics. 2001 Nov; 17(11): 1053-7.

Licensing Information

http://transpath.gbf.de

DIP

Description

The Database of Interacting Proteins (DIP) is a database that documents experimentally determined protein-protein interactions. This database is intended to provide the scientific community with a comprehensive and integrated tool for browsing and efficiently extracting information about protein interactions and interaction networks in biological processes.

URL

http://dip.doe-mbi.ucla.edu

References

Xenarios I, Rice DW, Salwinski L, Baron MK, Marcotte EM, and Eisenberg D. 2000. DIP: The Database of Interacting Proteins. Nucleic Acids Research 28:289-91.

Xenarios I, Fernandez E, Salwinski L, Duan XJ, Thompson MJ, Marcotte EM, and Eisenberg D. 2001. DIP: The Database of Interacting Proteins: 2001 update. Nucleic Acids Research 29:239-41.

Xenarios I, Salwinski L, Duan XJ, Higney P, Kim S, and Eisenberg D. 2002. DIP: The Database of Interacting Proteins. A Research Tool for Studying Cellular Networks of Protein Interactions. Nucleic Acids Research 30:303-5.

Licensing Information

Academic users may freely download DIP data. Registration is required at http://dip.doe-mbi.ucla.edu/dip/Login.cgi?R=1.

Non-academic users must obtain a license. For more information, see http://dip.doe-mbi.ucla.edu/dip/Login.cgi?R=1

Pre-Loaded Data

Metabolic Pathways

Glycolysis, Gluconeogenesis, and TCA Cycle

Lehninger AL, Nelson DL, and Cox MM. *Principles of Biochemistry*. Worth Publishing, 2000. Third Edition.

Pentose Phosphate (Pi) Pathway

Stryer L. *Biochemistry*. W.H. Freeman and Company, 1995. Fifth Edition.

Signal Transduction Pathways

EGF

Schoeberl B, Eichler-Jonsson C, Gilles ED, and Mueller G. 2002. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. Nature Biotechnology 20: 370-75.

TNFR

Chen G, Goeddel DV. 2002. TNF-R1 Signaling: A beautiful pathway. Science 296:1634-35.

Wnt

Moon RT, Bowerman B, Boutros M, and Perrimon N. The promise and perils of Wnt signaling through beta-catenin. 2002. Science 296:1644-46.

Gene Expression

The expression data described in *Chapter 8* was obtained from the article referenced below. A tab-delimited text file is included in the default database that is installed with Vector PathBlazer and is located in *C:\VNTI Database\PathwayDB\DeRisi_glycolysis_exp_import.txt*.

DeRisi JL, Iyer VR, and Brown PO. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278(5338):680-86.

Interaction Generality

Saito R, Suzuki H, and Hayashizaki Y. 2002. Interaction generality, a measurement to assess the reliability of a protein-protein interaction. Nucleic Acids Research 30(5):1163-68.

APPENDIX

D

TROUBLESHOOTING

This appendix contains a list of troubleshooting tips to aid in solving problems you might encounter when using Vector PathBlazer.

General

Problem: When a new component is created from one of the shapes in the Palette window and is named, a name other than the one entered displays in the Graphics window.

Solution: When a component is added, the program searches the existing components in the database by name and by synonym for a match. If the entered name is a synonym to an existing component then the primary name of that component displays. Right click on the object and select **Change Component Display Name**. In the dialog box that opens, select the display name from those listed. You can select an option to change the name in the current image or to change it in all pathways.

Problem: When trying to add a new component that does not already exist in the database by primary name or synonym, the program finds a match and names the component anyway. For example, when attempting to create a new component called bAR (beta ardrenergic receptor, involved in the G-protein signaling pathway), the program automatically names the object F-actin instead because ‘barbed’ is a synonym of F-actin.

Solution: The searching algorithm treats the component name as a string and finds all instances of the matching string partially or completely. The program automatically renames the component if there is only one match in the database. If there is more than one match, a list of matching components is presented from which you can determine whether any of the options are a “true” match. To resolve this, right click on the object and select **Change Component Display Name**. In the dialog box that opens, select the display name from those listed. You can select an option to change the name in the current image or to change it in all pathways.

Problem: A new component is drawn in the Graphics window and, to name it, **Component Properties** is selected from the shortcut menu. In the Component wizard, the **Select component from database radio button** is selected and 'glucose' is entered. You know D-Glucose is already in the database and that one of its synonyms is 'glucose' but Glucose is not returned.

Solution: When an object is named by this method, objects in the database are only searched by primary name and not by synonym. Name the object in the Graphics window first by either double-clicking and entering a name or selecting **Object Properties** from the shortcut menu and entering a name in the Name field in the Properties box. When the Component wizard opens, the search will be performed by primary name and by synonym.

Import

This section lists errors that may be encountered when importing data.

Problem: When a proprietary XML file does not contain the attributes **<list_of_substances>** and/or **<list_of_interactions>** the error in *Figure D.1* is generated.

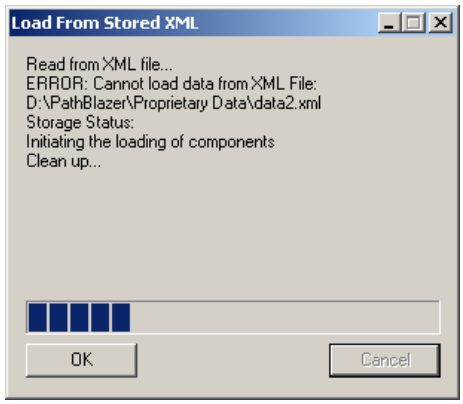


Figure D.1 Error generated when a required section is missing in the XML file

Solution: To import a proprietary XML file, both of the attributes **<list_of_substances>** and **<list_of_interactions>** must be present in the file, even if the attribute is empty. For example, the following file can be imported successfully even though both of the attributes **<list_of_substances>** and **<list_of_interactions>** are empty.

```
<storage ID="BIND:Storage">
 <list_of_substances>
</list_of_substances>
 <list_of_interactions>
</list_of_interactions>
</storage>
```

Problem: When an XML file does not define an attribute correctly in one of the entries after the first entry, the error in *Figure D.2* is generated. All entries that are defined correctly before the incorrect entry are imported into the database before the import halts. In the following example, a partial file is listed with the closing **</substance>** attribute crossed out to indicate it is missing, which would cause the error shown in *Figure D.2* . The error message also shows the last entry that was successfully loaded into the database.

```
...
<storage ID="BIND:Storage">
 <list_of_substances>
  <substance ID="Prostaglandin-E2 9-reductase">
   <list_of_origin_accesses>
    <origin_access>
     <database>KEGG</database>
     <access>EC 1.1.1.189</access>
     <item_URL>http://www.genome.ad.jp/dbget-bin/www_bget?ec:1.1.1.189</item_URL>
     <extra_data>KEGG Enzyme Link</extra_data>
    </origin_access>
</list_of_origin_accesses>
...
~~</substance>~~
</list_of_substances>
..
```

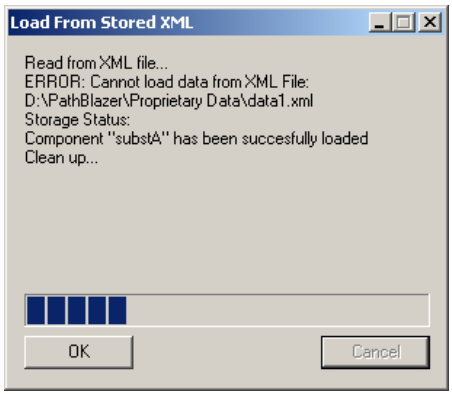


Figure D.2 Error generated when an attribute is defined incorrectly in the XML file

Solution: To determine where the source of the error is in the file, look for any incorrectly defined attributes in the entry after the last entry that was successfully loaded.

Problem: When a proprietary XML file does not define an attribute correctly in the first entry, the error in *Figure D.3* is generated. All entries that are defined correctly before the incorrect entry are imported into the database before the import halts. In the following example, a partial file is listed with the opening **<synonyms>** attribute crossed out to indicate it is missing, which would cause the error shown in *Figure D.3* .

```
<storage ID="BIND:Storage">
 <list_of_substances>
  <substance ID="Prostaglandin-E2 9-reductase">
   <list_of_origin_accesses>
    <origin_access>
     <database>KEGG</database>
     <access>EC 1.1.1.189</access>
     <item_URL>http://www.genome.ad.jp/dbget-bin/www_bget?ec:1.1.1.189</item_URL>
     <extra_data>KEGG Enzyme Link</extra_data>
    </origin_access>
   </list_of_origin_accesses>
   <synonyms>
    <name>Prostaglandin-E2 9-reductase</name>
      <name>(5Z,13E)-(15S)-9alpha,11alpha,15-Trihydroxyprosta-5,13-dienoate:NADP+  9-
oxidoreductase</name>
    <name>EC 1.1.1.189</name>
```

</synonyms>

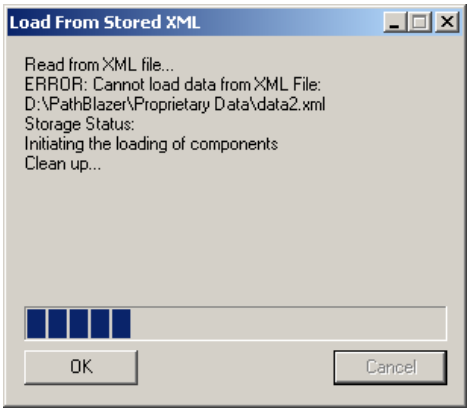


Figure D.3 Error generated when an attribute is incorrectly defined in the first entry

Solution: To determine where the source of the error is in the file, look for any incorrectly defined attributes in the first entry of the file.

Glossary

.mdb file: Vector PathBlazer database file.

.pw file: Vector PathBlazer “min-database” file that that stores individual pathways and associated reaction and component data.

Alternate View: A copy of an existing view or a new view in the Graphics window that is saved with a pathway. Components, connectors, or reactions in a pathway cannot be added or changed in an Alternate View but the graphical properties of the pathway elements and the graph can be changed.

Annotation: A descriptive property of a component, connector, reaction, or pathway such as name or cellular location.

Component: One of the main database object types that is an element of a reaction. Can be either an input or an output of the reaction and can be any kind of molecule such as protein, DNA, RNA, or small molecule. Can also be a physical element such as heat or light.

Connector: Secondary database object type that links a component to a reaction node. Can be unidirectional (forward or reverse), bidirectional (catalytic), or non-directional (protein-protein interaction).

Discovery View: Type of view where catalytic reactions (those that involve an enzyme and a bidirectional connector) are displayed as individual objects in a reaction. Protein-protein interactions with non-directional connectors can also be displayed.

Experiment or **Runs Project**: A collection of Expression Runs combined for simultaneous analysis.

Expression Run: In the context of Vector Xpression, an array of numbers (equal in length to the number of Expression Genes that were measured) that corresponds to the expression values obtained when an Expression Target is put through the measurement oprotocol (I.E. a microarray hybridization or SAGE run).

Interaction Generality: Number of proteins that directly interact with the target protein pair minus the number of proteins that interact with more than one protein plus one. A lower generality score indicates a more biologically relevant protein-protein interaction.

Label: Displays additional information or titles on a component, reaction node, or connector in the Graphics window.

Master View: Tab in the Graphics window in which a pathway and its associated data is viewed in graphical format as opposed to text format.

Metabolic View: Type of view where catalytic reactions (those that involve an enzyme and a bidirectional connector) are not displayed as individual objects in a reaction. Instead, an enzyme displays as a label of the reaction node and the connector does not display. Protein-protein interactions with non-directional connectors cannot be displayed in this type of view.

Non-strict Search: Search term that returns objects that are assigned the value of In or Known In for Location and Organism annotations. Also returns objects that are assigned no value for Location and Organism annotations.

Pathway: One of the main database object types that is made up of one or more reactions linked together. Different types of pathways can be modeled in Vector PathBlazer including metabolic and signal transduction pathways. Pathways can also be made up of networks of protein-protein interactions.

Pooling: Refers to displaying just one time a component that occurs more than once in a pathway. In the Graphics window, multiple connectors are drawn from the one object to the reactions in which it is involved.

Protein-Protein Interaction: Reaction between two proteins.

Reaction Node: Graphical representation of a reaction in the Graphics window.

Reaction: One of the main database object types that is made up of groups of one or more components that undergo a transformation or interaction.

Strict Search: Search term that returns objects that are assigned the value of In or Known In for Location and Organism annotations.

Subset: A type of container that contains references to objects in the database and can be used to group objects with one or more properties in common.

Synonym: Alternate name or alias of a component.

Template: Defines the mapping between a set of gene names from a gene expression data file and a set of component names.

Text View: Tab in the Graphics window in which a pathway and its associated data is viewed in text format as opposed to graphical format.

Index

A

Adding
- Alternate Views in Graphics window 29
- annotations to objects 37
- components to Graphics window 113, 118, 121
- component to saved reaction 128
- folders to Database Explorer 34
- labels to Graphics window 131
- molecules to commonly used molecules list 111
- reactions to Graphics window 122, 125
- reaction to Graphics window 125
- reverse reactions 35
- search results to subsets 59
- subsets to Database Explorer 34

Alternate View
- copying 31
- creating 30
- deleting 31
- description 29

Annotating
- objects, description 37
- objects as a batch 38
- objects with GO annotations 157

Annotations
- component 39
- connectors 44
- description 37
- pathway 39
- reaction 39

Annotations See Gene Ontology 153
Attribute See Annotations 37

B

Background color in Graphics window 20
Batch annotation 38
BIND
- description 80
- import instructions 84
- import logic 81, 83
- source files 81

BioCyc
- Component files 87
- description 85
- import instructions 92
- import logic 86
- Pathways file 90
- Reaction files 89
- source files 86

Border color in Graphics window 20
Browsing in Database Explorer 31
Building pathway
- adding a stepwise reaction 147
- default colors 138
- from starting component 139
- from starting pathway to ending component 143
- from starting to ending component 141
- link between two pathways 149
- showing connection from other datasources 150
- through a component 145

C

Circular layout
- applying 24
- description 24
- properties 25

Colors
- expression values in Graphics window 179
- in automatically assembled pathway 138
- modifying expression value display 179

Color schema
- applying universally 21
- creating 21

Commonly used molecules
- adding 112
- deleting 112
- description 111
- editing 112

Component
- adding to Graphics window 113
- annotation fields 39
- changing display name 113
- commonly used molecules 111
- copying 33
- deleting 33
- deleting from saved reactions 130
- description 6
- displaying database crosslinks 52
- hiding in Graphics window 19
- joining into reactions 122
- renaming 33
- viewing graphical properties 19
- viewing in Text View 29
- viewing properties 29

Components
- merging manually 45

Connector
- adding in Graphics window 122
- annotation fields 44
- changing in saved reactions 131
- deleting from saved reactions 131
- description 14
- direction 45
- hiding in Graphics window 19

joining components into reactions 122
navigating in Graphics window 18
viewing graphical properties 19
viewing in Text View 29
viewing properties 29
Copying Alternate Views 31
Creating
Alternate View 30
component subsets from reaction/pathway 35
database 10
empty subsets 34
folders 34
reaction subsets from pathway 35
subsets 34
subsets with contents 34
Crosslinks
defining annotation 41
opening from Graphics window 52
Customizing
column display in Database Explorer 32
gene ontology display 22
graphical layouts 24
graphical properties 19
universal color schemes for objects 21

D

Database
.mdb file 6
backing up 11
creating 10
default installation 6
description 6
main data types 6
pre-loaded data 7
selecting .mdb file for use 10
updating from PathBlazer 1.0 11
Database Explorer
adding components/reactions/pathways to Graphics window 36
browsing data 31
building pathways 139
changing column display 32
Contents Pane 13
creating folders 34
creating subsets 34
description 12
hiding 13
List Pane 13
moving 13
organizing data 33
reversing reaction direction 35
searching database 54
Database search
multiple conditions 54
single condition 54
Data Import See Importing 66
Data types
component 6
pathway 7
reaction 6
Deleting
Alternate View 31
annotations 116
component from saved reaction 130
components from commonly used molecules list 112
connectors 131
folders 34
labels 132
objects 33
subsets 35
Demo Mode 183
description 14
DIP
data display 100
description 97
import instructions 99
import logic 97
source files 97
Direction
connector annotation 45
in pathway building 137
of a reaction 7, 14
reversing for reaction 35
Discovery View
building a pathway 138
description 14, 109
opening new Graphics window 112
Displaying
Database Explorer 13
expression values on components 178
gene ontology annotations 22
Palette window 12
status bar 11
Document Type Description
see DTD 191
Drawing
existing component in Graphics window 118, 121
existing reaction in Graphics window 125
new component in Graphics window 113
new reaction in Graphics window 122
opening Graphics window 112
pathway in Discovery View 109
pathway in Metabolic View 109
tools 110
DTD
importing proprietary data in XML format 191
Dynamic License 183

E

Editing

expression template 170
GO annotations 159
Experiments definition 7
Expression data
Annotation field 44
assigning display colors to value ranges 179
bar graph values 179
creating template for import 166
displaying values on pathway components 178
importing with template 168
introduction 165
linking to pathway components 166
mapping database links manually 171
modifying expression value colors 179
sending to PathBlazer from Vector Xpression 177
Expression Palette 179

F

Filtering schema
creating 21
hiding 21
Folders
adding subfolders and subsets 34
creating 34
deleting 34
Font
changing in Graphics window 20, 21

G

Gene expression See Expression data 165
Gene ontologies
annotating objects 157
annotation field 43
downloading files 153
examples 162
importing 153
importing annotations 159
Gene Ontology
customizing display 22
deleting a GO annotation 159
editing a GO annotation 159
importing GO annotations 159
importing GO terms 154
introduction 153
linking to pertinent websites 157
searching database for objects with annotations 61
searching for GO terms 156
updating GO categories 159
viewing GO terms 155
Graphical layouts
applying 28
description 24
properties 25
types 24
Graphical properties
changing 20
customizing 19
viewing an object's 19
Graphics window
adding components from Database Explorer 36
adding labels 131
adding pathways from Database Explorer 36
adding reactions from Database Explorer 36
changing object's graphical properties 20
displaying expression values on components 178
drawing existing component 118, 121
drawing existing reaction 125
drawing new component 113
drawing new reaction 122
example reactions 15
finding objects in pathway 53
fitting image 18
hiding and unhiding objects 19
modifying graph properties 24
modifying saved reactions 127
navigating 17
opening 112
opening database crosslinks 52
opening pathways 14
panning an image 17
printing images 63
rearranging objects 17
resizing images 17
saving images 63
selecting objects 18
viewing pathways 13, 14
zooming 18
Graph properties
modifying 24

H

Hiding
columns in Database Explorer 32
Database Explorer 13
objects in Graphics window 19
Palette window 12
status bar 11
Hierarchical layout
applying 25
description 25
properties 26
Highlighting schema
creating 21
hiding 21

I

Imporing
TransPath 93
Importing
BIND 80

BioCyc 85
description of data import 65
DIP 97
DTD for importing proprietary data 191
expression data 168
expression template 171
gene ontologies 153
gene ontology terms 154
KEGG 72
PPI 100
pre-defined URLs 107
proprietary data 102
root folder description 67
session monitor 70
source file description 67
steps for general import 66
Import Manager
description 66
In description 56
Interaction generality
definition 137
setting for pathway building 137
Interactive Zoom 18
Intersection of subsets 34

K

KEGG
Compound file 73
description 72
Enzyme file 75
Genome file 78
import instructions 79
import logic 73
Reaction files 77
source files 72
Known In description 56

L

Label
changing graphical properties 132
creating 131
deleting 132
description 131
Launching PathBlazer Viewer 10
Layout
circular 25
dialog box 25
hierarchical 25
symmetrical 25
Layout properties 25
License Manager 183
configuring dynamic license 187
configuring trial license 187
resetting static 186
Licenses, Vector Xpression 183
License status 183
Linking
expression data and components manually 171
expression data to pathway components automatically 166
Log file
contents of 72
permanent 71
Logical conditions for database search 58

M

Manual conventions 4
Marquee Zoom 18
Master View
copying as Alternate View 31
creating new Alternate View 30
description 12
opening a pathway in 14
viewing pathways graphically 14
mdb file
backing up 11
creating new database 10
default database 6
description 6
opening in PathBlazer Viewer 14
selecting database for use 10
Merge Option dialog box 67
Merging components
criteria 68
description 45
during data import 67
manually 45
merge rules during import 68
Merging data, results 69
Metabolic View
building a pathway 138
description 14, 109
opening new Graphics window 112

N

Navigating
Database Explorer 31
Graphics window 17
Non-strict search
definition 56
Not In description 56

O

Object properties
viewing 19
Online Help 3
Ontology See Gene Ontology 153
Opening

components in Graphics window 36, 118, 121
pathways in Graphics window 14, 37
reactions in Graphics window 37, 125
Organism GO annotation 161
Organizing data 33
Overview window
resizing images in Graphics window 17
tiling with Palette window 18

P

Palette window
commonly used molecules 111
component shapes/connector links 110
description 12
hiding 12
moving 12
reanchoring in position 12
Panning in Graphics window 17
PathBlazer
interaction with Vector Xpression 166
main features 5
overview 5
tools opened from Vector Xpression 166
PathBlazer Viewer
elements 11
launching 10
opening a .mdb file 14
opening a .pw file 14
Pathway
annotation fields 39
browsing in Database Explorer 31
copying 33
default colors 138
definition 7
deleting 33
Discovery View description 14
displaying database crosslinks 52
graphical representation in Vector PathBlazer 14
Metabolic View description 14
opening in Graphics window 14
renaming 33
saving from file to database 51
viewing in Text View 28
viewing properties 28
Pathway building
adding stepwise reactions 138
assembly parameters 134
examples 139
excluding components 136
from Database Explorer 139
hiding components 136
in Discovery View 138
in Metabolic View 138
selecting reaction subset 134, 135
setting additional step number 136
setting interaction generality 137
setting maximum step number 136
specifying direction 137
specifying start/end/through component 135
turning pooling off 136
Pathway Viewing Area
elements 12
Master/Text Views 12
Pooling
definition 136
turning off for pathway building 136
Populating GO Organism annotation 161
Populating GO Subcellular Location annotation 161
PPI
data display 102
description 100
import instructions 100
Pre-loaded data
crosslinks to Vector Advance 52
description 7
in gene expression display 175
Printing
images/text from Graphics window 63
Properties
Component Class tab 43
Component tab 39
Condition tab 44
Constants tab 44
Cross Links tab 41
Expression Data tab 44
General tab 39
GO Annotations tab 43
graphical 19
graphical layout 25
in Database Explorer 31
in Text View 29
Locations tab 42
object 19
Organisms tab 40
Pathway tab 44
References tab 43
Synonyms tab 43
viewing annotations 37
Proprietary data
defining components 102
defining components in import 102
defining pathways in import 104
defining reactions in import 103
description 102
import 102
import instructions 105
Protein-protein interaction
definition 15
Protein-protein interactions
building pathways 137
drawing 109

viewing 14
pw file
description 14
opening in PathBlazer Viewer 14, 113
saving to database 51

R

Reaction
annotation fields 39
copying 33
definition 6
deleting 33
direction 14
drawing in Graphics window 122
hiding in Graphics window 19
modifying saved reactions 127
properties 29
renaming 33
reversing direction 35
viewing in Text View 29
Reaction node
definition 14
viewing graphical properties 19
Reactions, saving 50
Rearranging objects in Graphics window 17
Renaming component display name 113
Resizing images in Graphics window 17

S

Saving
.pw file to database 51
components 114, 119
image formats 63
images from Graphics window 63
pathway to .pw file 46
pathway to database 46
reactions separate from pathway 50
reactions to pathway 46
Search
setting logical conditions 58
Searching
adding results to existing subsets 59
adding results to new subsets 60
configuring conditions 55, 57
database for GO annotated objects 61
for objects in database 54
for objects in pathway 53
Non-Strict Search 56
results display 58
Strict Search 56
Vector Xpression database 176
with multiple conditions 54
with single condition 54
Search Results display 58
Selecting
objects in Database Explorer 13
objects in Graphics window 18
Status bar hiding 11
Stepwise reaction
adding from Graphics window 125
in pathway building 138
Strict search definition 56
Subcellular Location GO annotation 161
Subsets
adding components to 130
adding reactions to 130
adding search results 59, 60
creating 34, 130
creating component subsets from reaction/pathway 35
creating intersection of 34
creating reaction subsets from pathway 35
creating union of 34
deleting 35
selecting for pathway building 134
Symmetric layout
applying 25
description 25
properties 27
Synonym
adding to component 117
annotation field 43
definition 6
drawing a new component 113
System requirements 2

T

Tab-delimited file for expression values 174
Technical Support 4
Template
creating expression 166
creating template from Vector Xpression 176
editing a 170
editing expression 170
importing expression 171
importing expression data with 168
Text View
Components folder 29
description 12, 28
Pathway folder 28
Reaction folder 29
TransPath
auxiliary files 95
custom dictionaries 95
description 93
import instructions 96
source files 93
Trial License 183
Troubleshooting
general problems 203

import problems 204

U

Unhiding objects in Graphics window 19
Union of subsets 34
Updating database 11
URLs predefined 107

V

Vector Xpression
creating template from for expression data import 176
finding components in PathBlazer from 177
interaction with PathBlazer 166
licenses 183
opening Experiment in from PathBlazer 177
searching database from PathBlazer 176
sending expression data to PathBlazer 177
tools opened from PathBlazer 166
Viewing
object's graphical properties 19
pathways in Graphics window 13, 14
pathways in text format 28

Z

Zoom
features in PathBlazer 18
fitting image 18
in 18
Marquee 18
out 18
Zooming
Interactive 18